

FEASIBILITY STUDY OF AN
INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE
DESIGN (IPAD) SYSTEM

by C. A. Garrocq, M. J. Hurley, M. Dublin, et al

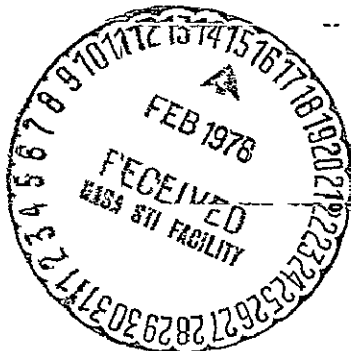
VOLUME VI

IMPLEMENTATION SCHEDULE
DEVELOPMENT COSTS
OPERATIONAL COSTS
BENEFIT ASSESSMENT
IMPACT ON COMPANY ORGANIZATION
SPIN-OFF ASSESSMENT

(PHASE II, TASKS 3 to 8)

30 August 1973

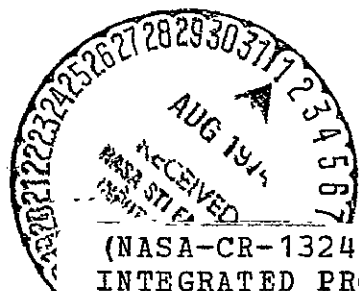
Publicly Released
February 10, 1978



Prepared Under Contract No. NAS 1-11431 by

GENERAL DYNAMICS/CONVAIR AEROSPACE DIVISION
San Diego, California
for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION



(NASA-CR-132406) FEASIBILITY STUDY OF AN
INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE
DESIGN (IPAD) SYSTEM. VOLUME 6:
IMPLEMENTATION SCHEDULE, DEVELOPMENT COSTS,
OPERATIONAL (General Dynamics/Convair)

N78-16009

HC A08/MF A01

Unclas

G3/02 02563

FEASIBILITY STUDY OF AN
INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE
DESIGN (IPAD) SYSTEM

- VOLUME I - SUMMARY
- VOLUME II - CHARACTERIZATION OF THE IPAD SYSTEM
(PHASE I, TASK 1)
- VOLUME III - ENGINEERING CREATIVE/EVALUATION PROCESSES
(PHASE I, TASK 1)
- VOLUME IV - DESIGN OF THE IPAD SYSTEM
PART I - IPAD SYSTEM DESIGN REQUIREMENTS
(PHASE I, TASK 2)
- VOLUME V - DESIGN OF THE IPAD SYSTEM
PART II - SYSTEM DESIGN
PART III - GENERAL PURPOSE UTILITIES
(PHASE I, TASK 2)
- | | |
|-----------|--|
| VOLUME VI | - IMPLEMENTATION SCHEDULE
DEVELOPMENT COSTS
OPERATIONAL COSTS
BENEFIT ASSESSMENT
IMPACT ON COMPANY ORGANIZATION
SPIN-OFF ASSESSMENT
(PHASE II, TASKS 3 to 8) |
|-----------|--|

FOREWORD

This investigation was conducted for the NASA Langley Research Center by the Convair Aerospace Division of General Dynamics Corporation under Contract NAS 1-11431.

The NASA Technical Monitor was Dr. Robert E. Fulton, Head, IPAD Technology Section, Design Technology Branch, Structures and Dynamics Division, assisted by Dr. Jaroslaw Sobieszczanski and Mrs. Susan J. Voigt. The Convair Project Leader was Mr. C. A. Garrocq, assisted by Messrs. M. J. Hurley, M. Dublin, J. J. Hosek, O. E. Wheeler, R. A. Westerwick, R. H. Trelease, J. S. Barron, G. W. Bradfield, J. D. Sutliff, W. J. Moran, A. Karemaa, R. T. Schemensky, L. A. Evans, D. S. Osterle, R. G. Huntington, J. L. Howse, H. Roland, R. W. McMickle, K. C. Bonine, J. E. Meyer, R. H. Schappelle, D. Kurpiewski, S. K. Pederson, R. Howe, N. E. Tipton, A. H. Cooper, G. M. Oing, J. M. Maughmer, W. E. Jenkins, T. M. Wooster, J. C. Mallory, B. H. Oman, D. W. Peterson, P. K. Shih, D. R. Diaddigo, R. E. Kenyon, B. J. Kuchta, J. D. Neilson, J. T. Gordon, and W. H. Buckner.

The Control Data Corporation participated in the performance of this study as a subcontractor to the General Dynamics Corporation.

The period of performance was from 15 March 1972 to 30 August 1973.

TABLE OF CONTENTS

Section		Page
1	INTRODUCTION	1
2	IPAD SYSTEM IMPLEMENTATION PLAN	5
2.1	Assumptions and Groundrules	7
2.2	Baseline Schedule and Cost Summary Chart	8
2.3	Network Planning Technique Used	10
2.4	Phase I - Task 1, Requirements	13
2.4.1	Operational module (OM) specifications	13
2.4.2	Project management software refurbishing specifications	17
2.4.3	EXECutive specifications	22
2.4.4	General graphics library (GGL) specifications	25
2.4.5	IPAD system requirements	29
2.4.6	FORTRAN temporary code (FTC), FORTRAN data description language (DDL) and data manipulation language (DML) specifications	34
2.4.7	Device/media control language (DMCL)	45
2.4.8	General purpose utility (GPU) specifications	47
2.4.9	Special purpose utility (SPU) specifications	48
2.5	Phase I - Task 2, Development	48
2.5.1	Project management capability	48
2.5.2	Project-oriented management/engineering software and nonexecutable Code No. 1	52
2.5.3	EXECutive program for computing system No. 1	58
2.5.4	Data base management system (DBMS)	64
2.5.5	General graphics library (GGL)	69
2.5.6	General purpose utilities (GPU) and stand- alone general design module (GDM)	72
2.5.7	Special purpose utilities (SPU)	80
2.6	Phase I - Task 3, IPAD First Release Capability System No. 1	85
2.6.1	Management/engineering OM and SPU subsystem checkout, Block 75	89
2.6.2	Design OM and GPU subsystem checkout, Block 76	90

TABLE OF CONTENTS, Contd

Section	Page
2.6.3 IPAD system No. 1 checkout, Block 77	90
2.6.4 IPAD system no. 1 demonstration, Block 78	90
2.7 Phase I - Implementation Plan Management	91
2.7.1 Management and coordination, IPAD system No. 1	91
2.7.2 Coordination meetings	91
2.8 Reporting	92
2.8.1 Oral presentations, Blocks 81 and 82	92
2.8.2 Monthly technical and management reports, Block 83	92
2.8.3 Final report, Blocks 84 to 88	92
2.9 Phase II - Deployment to Computing Systems No. 2 and No. 3	92
2.9.1 Main contractor tasks	93
2.9.2 IPAD system No. 2, implementation plan	95
2.9.3 IPAD system No. 3 implementation plan	98
2.10 Alternative Approaches	98
2.10.1 GGL language development	99
2.10.2 FORTRAN DML and DDL language development	101
2.10.3 GDM improvements	102
2.11 Cost Distribution Curves	104
2.12 Variants to Reduce Peak Funding and Costs	106
2.12.1 Variant to reduce peak funding	106
2.12.2 Alternatives designed to reduce costs	107
 3 IPAD SYSTEM OPERATIONAL COSTS (PHASE II, TASK 5)	 109
3.1 Identifiable Cost Increases	111
3.1.1 EXEC overhead	111
3.1.2 DBMS overhead	111
3.1.3 Costs associated with increased host computer hardware	113
3.1.4 Terminal Costs	113
3.1.5 Conclusions	113
3.2 Identifiable Cost Decreases	114
3.3 Summary	115

TABLE OF CONTENTS, Contd

Section		Page
4	IPAD SYSTEM BENEFIT ASSESSMENT (PHASE II, TASK 6)	117
4.1	Impact of IPAD on Performance of Aerospace Vehicles	118
4.1.1	STOL tactical aircraft study	118
4.1.2	Navy V/STOL fighter attack aircraft	121
4.1.3	Space shuttle launch study	122
4.1.4	Large space telescope study	124
4.1.5	Overview	125
4.2	Impact of IPAD on Engineering Work	125
4.3	Disciplinary Interfaces	128
4.4	Answers to NASA's RFP Questions	129
5	IPAD SYSTEM IMPACT ON COMPANY ORGANIZATION (PHASE II, TASK 7)	131
5.1	Ramifications of Organizations Relative to Acceptance of IPAD	131
5.2	Change in Design Organizations to Suit IPAD	139
5.3	Answers to NASA RFP Questions	140
6	IPAD SPIN OFF ASSESSMENT (PHASE II, TASK 8)	143
6.1	Spin-Off in Technical Non-Aerospace Fields	143
6.2	Spin-Off in Business Non-Aerospace Fields	146
6.3	Answers to NASA's RFP Questions	151
	REFERENCES	153

LIST OF FIGURES

Figure		Page
1-1	Flow Chart of IPAD Study	2
1-2	Major IPAD Features	4
2-1	Major IPAD System Components	6
2-2	Computer Software Associated with IPAD	6
2-3	Baseline Implementation Plan Schedule and Control Summary	9
2-4	Variant of the CPM Task Block	10
2-5	CPM Network Symbols	12
2-6	IPAD Implementation Plan - Task 1, Requirements	14
2-7	Development Phase Project Management Capability	49
2-8	Development Phase - Project-Oriented Management/ Engineering Software	53
2-9	Development Phase - EXECutive Program for Computing System No. 1	58
2-10	Development Phase - Data Base Management System	64
2-11	Development Phase - GGLs, GPUs, and Stand-Alone GDM	70
2-12	Development Phase - Special Purpose Utilities (SPUs)	80
2-13	Phase I - Task 3, FRC Checkout and Demonstration	86
2-14	Phase I - Management and Reporting	91
2-15	Phase II, Main Contractor Tasks	93
2-16	Phase II - IPAD System No. 2 Implentation Plan (Identical for System No. 3 Except for Dates)	95
2-17	IPAD Implementation Plan, Phase II - Checkout and Demonstration System No. 2 (Identical for System No. 3 Except for Dates)	96
2-18	Phase II - Management and Reporting, System No. 2 (Typical for System No. 3)	96
2-19	Alternative GGL Development	99
2-20	Alternative DML and DDL Language Development	101
2-21	GDM Improvement Plan	102
2-22	Cost Distribution Curves for Man-Months and Computer Hours	105
2-23	Six-Month Stretchout Variant to IPAD Implementation Plan	106
3-1	Options Available to Use Saved Time	110
3-2	I/O Event Analysis Utilizing CDC's DBMS Design	112
3-3	Monitor Request per Case	112
4-1	Configuration Design Tradeoffs	119
4-2	Performance Design Tradeoffs	120
4-3	EBF Bypass Ratio Tradeoff	120

LIST OF FIGURES, Contd

Figure		Page
4-4	Space Shuttle Engine Deflection-Time History	123
5-1	Functional Organization Example Chart	133
5-2	Project Organization Example Chart	134
5-3	Division Manuals Relationships	135
5-4	VTOL Aircraft Project Organization	138

LIST OF TABLES

Table		Page
2-1	IPAD Baseline Implementation Plan, Costs Per Fiscal Year	104
2-2	6-Month Stretchout IPAD Implementation Plan Break-down Per Fiscal Year	106
2-3	IPAD Development Alternatives to Reduce Costs	107
4-1	STOL Tactical Aircraft Early and Final Design Comparisons	119
4-2	Comparison of Batch Mode and Interactive Mode Operations, STOL Tactical Aircraft	121
4-3	Comparison of Batch Mode and Interactive Mode Operations, Space Shuttle Launch	124
4-4	Comparison of Batch Mode and Interactive Mode Operations, Large Space Telescope	124

SUMMARY

A Baseline Implementation Plan, including alternative implementation approaches for critical software elements and variants to the plan, was developed in Tasks 3 and 4. The basic philosophy of this plan was aimed at: 1) a progressive release of capability for three major computing systems, 2) an end product that was a working tool, 3) giving participation to industry, government agencies, and universities, and 4) emphasizing the development of critical elements of the IPAD framework software. The results of these tasks indicate an IPAD first release capability 45 months after go-ahead, a five-year total implementation schedule, and a total developmental cost of 2027 man-months and 1074 computer hours.

Several areas of operational cost increases and decreases were identified in relation to Task 5. Cost increases are mainly due to the impact of additional equipment needed and additional computer overhead for the EXECutive and the Data Base Management System. These costs are very dependent on the computing installation and the charging algorithm. Operational cost decreases are associated mainly with lower numbers of engineering man-hours to perform a task and more efficient management under IPAD due to greater visibility of project data and quicker reaction time in the control of program status.

The benefits of an IPAD system relate mainly to potential savings in engineering man-hours, reduction of design-cycle calendar time, and indirect upgrading of product quality and performance.

The impact of incorporating IPAD within a company organization has been found to be minimal, consisting mainly of the addition of a Data Base Administrator to a project structure, and the training of the engineering personnel on the use and exploitation of IPAD features. The use of IPAD within an organization requires full backing from high levels of management.

It is believed that an IPAD system will quickly spin off to many other technical and non-aerospace fields, in particular those whose tasks involve the handling of large volumes of data and participation of various disciplines.

ALGOL	ALGORITHMIC LANGUAGE
A/N	ALPHANUMERIC
ANSI	AMERICAN NATIONAL STANDARDS INSTITUTE
AREA	A CODASYL CONCEPT
BASIC	BEGINNER'S ALL-PURPOSE SYMBOLIC INSTRUCTION CODE
BATCH	BATCH MODE PROCESSING
CAD	COMPUTER AIDED DESIGN
CAM	COMPUTER AIDED MANUFACTURING
CM	CENTRAL MEMORY
COBOL	COMMON BUSINESS ORIENTED LANGUAGE
CODASYL	CONFERENCE ON DATA SYSTEMS LANGUAGES
CP	CENTRAL PROCESSOR
CPM	CRITICAL PATH METHOD
CPU	CENTRAL PROCESSOR UNIT
CRT	CATHODE RAY TUBE
DBA	DATA BANK ADMINISTRATOR
DBCTG	DATA BASE CONCEPT TASK GROUP (CODASYL)
DBLTG	DATA BASE LANGUAGE TASK GROUP (CODASYL)
DBMS	DATA BASE MANAGEMENT SYSTEM
DBTG	DATA BASE TASK GROUP (CODASYL)
DDL	DATA DESCRIPTION LANGUAGE
DDLC	DATA DESCRIPTION LANGUAGE COMMITTEE (CODASYL)
DLF	DISCIPLINARY LIBRARY FILE
DMC	SEE DMCL
DMCL	DEVICE MEDIA CONTROL LANGUAGE
DML	DATA MANIPULATION LANGUAGE
DOD	DEPARTMENT OF DEFENSE
DVST	DIRECT VIEW STORAGE TUBE
EBF	EXTERNALLY BLOWN FLAP
ERB	ENGINEERING REVIEW BOARD
ERBC	ENGINEERING REVIEW BOARD COORDINATOR
EXEC	IPAD EXECUTIVE
EXPANDER	EXPANDER INTERACTIVE UTILITY
FDBLTG	FORTRAN DATA BASE LANGUAGE TASK GROUP
FORTRAN	FORMULA TRANSLATING SYSTEM
FRC	FIRST RELEASE CAPABILITY (IPAD'S)
FTC	FORTRAN TEMPORARY CODE
GDM	GENERAL DESIGN MODULE
GGL	GENERAL GRAPHICS LIBRARY
GGP	GENERAL GRAPHICS PLOTTER
GPGT	GENERAL PURPOSE GRAPHICS TERMINAL
GPU	GENERAL PURPOSE UTILITY

PRECEDING PAGE BLANK NOT FILMED

GLOSSARY OF IPAD ACRONYMS (Cont'd.)

ICL	IPAD CONTROL LANGUAGE
IDEF	SHORT FOR INPUT DEFINITION
I/O	INPUT/OUTPUT
I/ODEF	THE COMBINATION OF THE IDEF AND ODEF OF AN OM
IPAD	INTEGRATED PROGRAM FOR AEROSPACE-VEHICLE DE
IRAD	INDEPENDENT RESEARCH:& DEVELOPMENT
IS&R	INFORMATION STORAGE AND RETRIEVAL
JOD	JOURNAL OF DEVELOPMENT
JOVIAL	JULES' OWN VERSION OF THE INTERNATIONAL ALGEBRAIC LANGUAGE
MACRO	LARGE OR OF THE HIGHEST ORDER
MAXI	LARGE SCIENTIFIC COMPUTER SYSTEM
MDB	MULTIDISCIPLINARY DATA BANK
MDBU	MDB UPDATE FILE
MENU	A TABLEAU OR LIST OF ITEMS
MICRO	SMALL OR OF THE LOWEST ORDER
MINI	SMALL COMPUTER
MIS	MANAGEMENT INFORMATION SYSTEM
NASA	NATIONAL AERONAUTICS AND SPACE ADMINISTRATIO
NASTRAN	NASA STRUCTURAL ANALYZER
ODEF	SHORT FOR OUTPUT DEFINITION
OM	OPERATIONAL MODULE
OPTUM	IPAD OPTIMIZER GENERAL PURPOSE UTILITY
OS	OPERATING SYSTEM
OSCL	OPERATING SYSTEM CONTROL LANGUAGE
PDB	PROJECT DATA BANK
PERT	PROGRAM EVALUATION REVIEW TECHNIQUE
PLC	PROGRAMMING LANGUAGE COMMITTEE (CODASYL)
PP	PERIPHERAL PROCESSOR
PRC	PRE RELEASE CAPABILITY (IPAD'S)
PRF	PROJECT REVIEW FILE
PS	PROJECT SCHEMA
QP	QUERY PROCESSOR
QPL	QUERY PROCESSOR LANGUAGE
QPS	QUERY PROCESSOR SESSION
QPSS	QUERY PROCESSOR SESSION SKELETON
RECORD	A CODASYL CONCEPT
RFP	REQUEST FOR PROPOSAL
SCHEMA	A CODASYL CONCEPT
SET	A CODASYL CONCEPT
SPU	SPECIAL PURPOSE UTILITIES

GLOSSARY OF IPAD ACRONYMS (Cont'd.)

STATUM	THE IPAD STATISTICAL UTILITY MODULE
STOL	SHORT TAKE OFF AND LANDING
SUBSCHEMA	A CODASYL CONCEPT
TAS	TUTORIAL AIDS SUPPORT
TCS	TASK CONTROL SEQUENCE
TCSS	TASK CONTROL SEQUENCE SKELETON
TOGW	TAKE OFF GROSS WEIGHT
TSA	TASK STATUS/ACTION FILE
TTY	TELETYPEWRITER
UF	USERS FILE
USER	ANY IPAD USER
UTT	USER TASK TRAJECTORY
V/STOL	VERTICAL/SHORT TAKE OFF AND LANDING
VTOL	VERTICAL TAKE OFF AND LANDING

1 INTRODUCTION

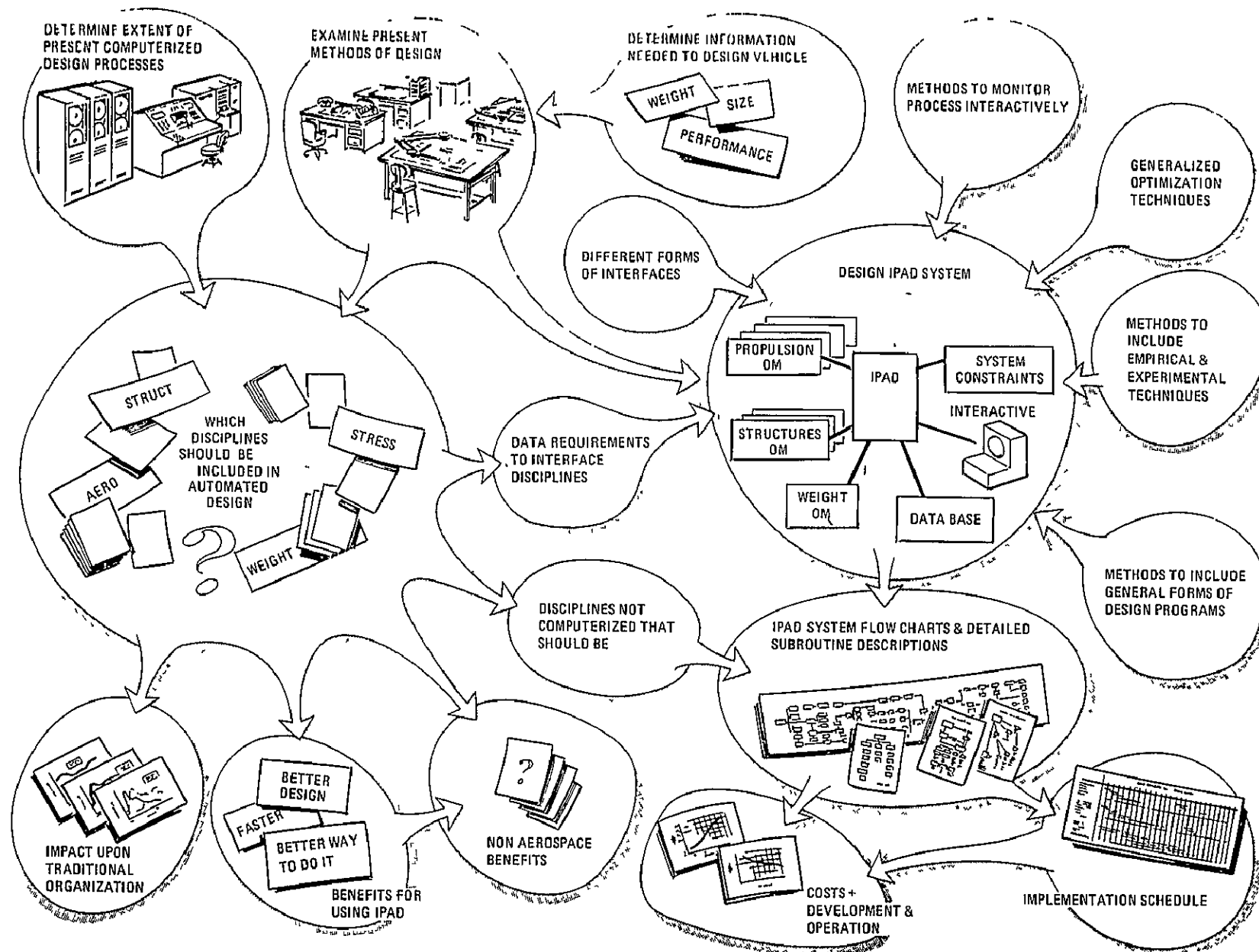
The design of a new aerospace vehicle is presently a complex, long-term process. At the onset, a set of objectives as identified in the areas of weight, performance, payload, range, etc., which are specified with a fairly good knowledge of the available design technology and constraints. The designer's goal is to minimize cost, while meeting basic project objectives. The designer possesses a fund of accumulated experience and knowledge which he applies, with intuition, to the requirements and constraints he has been given. The knowledge and experience of the designer are more and more frequently being delegated to the computer; the intuition and imagination can never be. Some of the purposes of the IPAD feasibility study were to determine what sections of the designer's tasks are amenable to automation; how much monitoring must the automation have; how can the design process be effectively organized; and, most important, how can the designer retain the visibility and control necessary to exercise his intuition and imagination in the design process.

The introduction of automation is a significant change in the design process, however, the important management aspects of this change are not only related to the technical details of engineering disciplines, programming, data bases, etc., but the key to success also depends upon managing the adaptation required of the people involved in the use of the automated process.

Automation of any process requires not only a thorough knowledge of the process, but of the pivotal factors that drive and control that process. When the process involves the myriad details of many programs and subroutines, thousands of variables, and the ramifications of computer operating system minutiae and coding "techniques," it is easy to lose sight of the fact that it is still the designer — the engineer — who is the key driver and decision maker in the process.

Although the various volumes of this report describe some of the considerations necessary for the technical basis needed to successfully automate the design process, the underlying, guiding philosophy has been that of providing a tool adapted to the needs of the designer — the ultimate user — and that is truly a useful tool. The acknowledged principle has been that the engineer and his management are generally more interested in solving the design problem than in becoming a better communicator with the computer.

The scope of the total IPAD feasibility study is illustrated in Figure 1-1. The study was divided into the following eight tasks within two study phases:



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 1-1. Flow Chart of IPAD Study

PHASE I

STUDY PLAN COORDINATION

TASK 1 - CHARACTERIZATION OF IPAD SYSTEM

- Define an IPAD Engineering Usage Philosophy
- Identify Feasible Automated Design Procedures
- Evaluate Adequacy of Existing Computer Programs
- Recommend Areas for Further Development
- Determine IPAD Feasibility and Applicability
- Recommend IPAD's First Release Engineering Capability

TASK 2 - DESIGN OF IPAD SYSTEM

- Define a Systems Operating Philosophy
- Evaluate System Design Options
- Identify Elements of IPAD's Utility Library
- Investigate Organization and Management of Data Bank
- Determine Number and Type of Input/Output Terminals
- Determine Host Computer Complex Configurations Adequate for IPAD
- Recommend IPAD's First Release Computer System Capability

PHASE II

TASK 3 - IPAD IMPLEMENTATION SCHEDULE

TASK 4 - IPAD SYSTEM DEVELOPMENT COST

TASK 5 - IPAD SYSTEM OPERATIONAL COST

TASK 6 - IPAD SYSTEM BENEFIT ASSESSMENT

TASK 7 - IPAD IMPACT ON COMPANY ORGANIZATION

TASK 8 - IPAD SPIN-OFF ASSESSMENT

Figure 1-2 summarizes the main features of an IPAD system as presently conceived and described elsewhere in this report.

IPAD IS

- AN INTEGRATED SYSTEM OF AUTOMATED MODULES.
EACH DISCIPLINE IS RESPONSIBLE FOR ITS OWN CAPABILITY DEVELOPMENT, UPDATE & GROWTH
- A USER-ORIENTED & DIRECTED MODULAR SYSTEM WITH FLEXIBILITY FOR CHANGE, ADAPTATION & EXPANSION
- A HARDWARE/SOFTWARE COMPUTER SYSTEM DESIGN APPROACH TO PERFORM ENGINEERING DESIGN PROCESSES MORE EFFECTIVELY, ECONOMICALLY & SWIFTLY
- A COMPUTER SYSTEM STRUCTURE
USABLE IN MANY ENGINEERING & SCIENTIFIC FIELDS
- ITS DATA BANK IS THE REPOSITORY FOR ALL DESCRIPTIVE & INFORMATIVE DATA GENERATED BY THE ENGINEERING/SCIENTIFIC TEAM FOR A SPECIFIC PROJECT
- A MANAGEMENT TOOL
TO PROVIDE IMMEDIATE VISIBILITY INTO PRODUCT STATUS & PROGRESS
- INITIALLY, A REASONABLE ENGINEERING CAPABILITY (SET OF AUTOMATED MODULES) MOUNTED ON A STATE OF THE ART HARDWARE/SOFTWARE STRUCTURE THAT CAN BE READILY IMPLEMENTED
- ULTIMATELY, A COMPREHENSIVE, DYNAMIC ENGINEERING TOOL SUPPORTED BY EFFICIENT, COST-EFFECTIVE HARDWARE/SOFTWARE CAPABILITY
- AN EDUCATIONAL AID FOR TRAINING NEW ENGINEERS IN THE USE OF VARIOUS DESIGN PROCESSES

IPAD IS NOT

- A SINGLE, HARDWIRED COMPUTER PROGRAM
- AN AUTOMATED, SINGLE-PURPOSE PROCEDURE
- A DISLOCATED ARRAY OF RANDOMLY COLLECTED COMPUTER PROGRAMS
- A SYSTEM OF PROGRAMS TO BE RUN BY A SINGLE DISCIPLINE
- A SYSTEM OF PROGRAMS IMPOSED BY AN AGENCY (OR COMPANY) ON THE AEROSPACE INDUSTRY COMMUNITY

Figure 1-2. Major IPAD Features

2 IPAD SYSTEM IMPLEMENTATION PLAN

This chapter presents the results of Task 3, IPAD Implementation Schedule, and Task 4, IPAD System Development Cost, of the IPAD Feasibility Study reported elsewhere in this report.

Implementation schedule and costs discussed herein refer to the IPAD System components shown in the three left-most boxes in Figure 2-1. A breakdown of these components into elements and subelements of software is shown in Figure 2-2, which is also a summary of the pieces of software required for IPAD (as identified in previous volumes of this report).

The general philosophy used in preparing this implementation plan can be summarized as follows.

1. The implementation of all major software components shall be blended in a progressive release, which evolves from demonstration cases into a prerelease capability (PRC) of individual subsystems, a first release capability (FRC) for Operating System No. 1, and, finally, deployment of IPAD to other operating systems.
2. The FRC shall be a properly checked-out and demonstrated working tool with management and engineering capability for comprehensive design and evaluation studies of an aerospace-vehicle project.
3. Competitive and sole-source participation shall be given to industry and institutions of higher education with the objective of bringing the best available know-how to bear on the design and implementation of IPAD.
4. The major IPAD contractor and integrator shall implement IPAD in one operating system (herein designated System No. 1) first and two other major aerospace companies shall implement IPAD in Operating Systems No. 2 and No. 3, respectively. The deployment of IPAD to other operating systems could be made in parallel or sequentially with this effort, but is not included in the baseline implementation plan.
5. The emphasis shall be in the development of the new IPAD framework and operating system software rather than in the development of new management/engineering capability. Substantial automated capability exists in the latter, with exception of configuration and subsystem design/drafting,

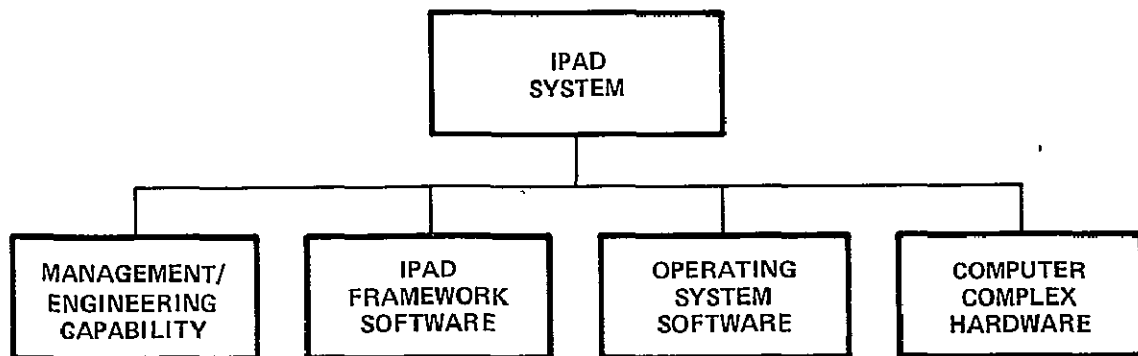


Figure 2-1. Major IPAD System Components

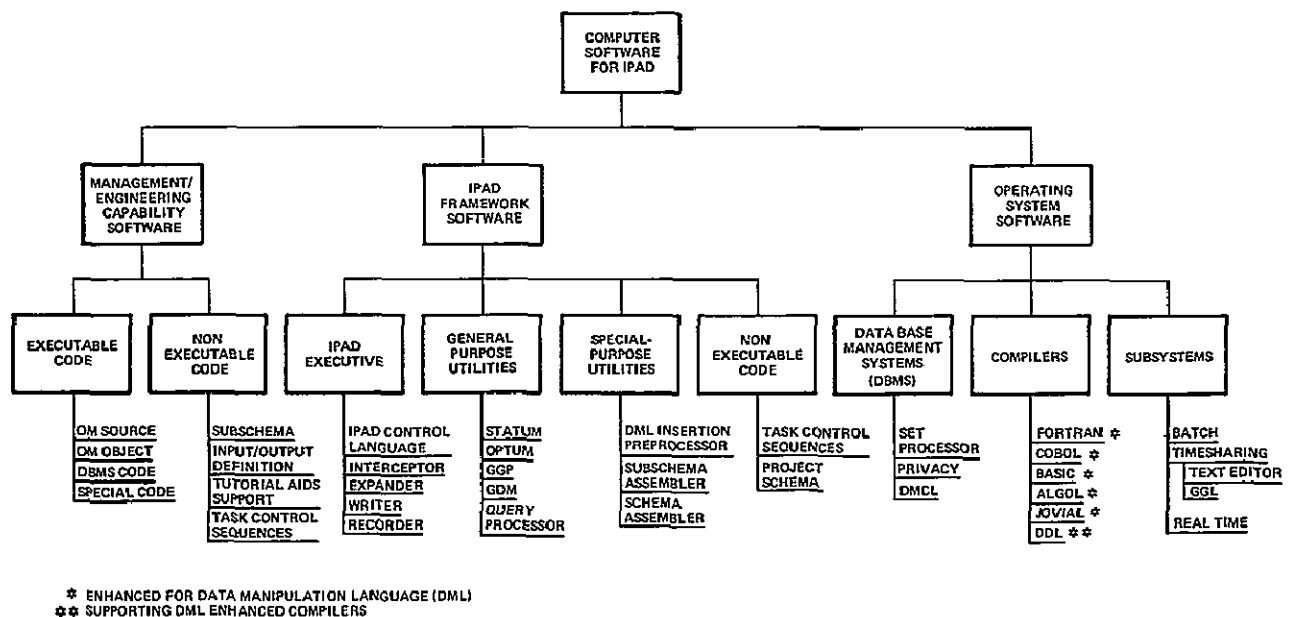


Figure 2-2. Computer Software Associated with IPAD

which needs only refurbishing to absorb the various new features of the IPAD system. The total software package for IPAD demonstration, however, shall contain all required elements properly interfaced to enable an engineering project team to perform a comprehensive aerospace-vehicle design and evaluation study.

In the discussions that follow, it is assumed that the reader has a general familiarity with the contents of Volumes II to V and, in particular, the postulated design of the IPAD system for which the implementation plan was derived.

2.1 Assumptions and Groundrules

Several general assumptions were made at the onset of Tasks 3 and 4 concerning the time frame, funding, and desirable features for IPAD development. These included:

1. Go-ahead was assumed to be in fiscal year (FY) 1975.
2. FY 1974 funds will be made available to support selected key pre-IPAD technology developments that represent critical items the implementation plan cannot realistically ignore. An alternative plan must be provided for the FY 1974 tasks in case these are not approved.
3. The IPAD implementation plan must produce a PRC as soon as possible and a functional FRC within four years from go-ahead.
4. The implementation schedule must provide convenient demarcations for systems review prior to commitment of a large block of funds.

In reference to both schedules and costs:

5. The implementation plan must include a visible risk assessment.

With respect to the end product to be implemented:

6. With due regard for cost and risk, the best designer/fabricator must be selected for any elements of software that can logically and realistically be developed separately.
7. It is a primary objective to bring about software which is both portable and "self maintaining" to the extent practicable. Codes supplied by each

computer manufacturer (e.g., an ANSI standard FORTRAN compiler) can be considered to meet this objective. Each has the capability represented by its code module, so portability is not an issue; since the manufacturer develops and maintains it, it is - in this regard - self maintaining. Further, code that exploits the host computer's operating system through well established interfaces is self-maintaining in that these interfaces typically survive system upgrades. However, the portable-code objective will be waived wherever the code is required to interface closely with the host operating system or wherever maximum efficiency is required. In this case, a code module will be tailored to a specific computing system, resulting in a functionally identical code module for each computing system accepting IPAD.

8. Development of new management/engineering software shall continue to be funded from conventional technology development studies at large and shall not be included in the development costs for IPAD.

2.2 Baseline Schedule and Cost Summary Chart

Figure 2-3 summarizes schedules and costs for the Baseline Implementation Plan. Detailed backup data for the various lines of this chart is given in Sections 2.3 to 2.9. Alternative approaches to the development of critical elements of software are discussed in Section 10, cost distribution curves are presented in Section 2-11, and variants to the Baseline Implementation Plan are presented in Section 2-12.

As indicated in Figure 2-3, the Baseline Implementation Plan is divided in two phases. Phase I is concerned with developing IPAD for a single computer operating system, but including portability to all three major operating systems for many software elements as shown by asterisks. Phase I culminates with a demonstration of the FRC for Operating System No. 1 in 45 months from go-ahead and final documentation at 50 months. Development costs of Phase I are estimated at 1385 man-months and 670 computer hours. Phase I is performed under a major IPAD contractor and integrator with subcontracted studies to industry and other institutions on a competitive or sole-source basis as discussed earlier.

Phase II is for the deployment of IPAD to two other major operating systems, using all common and portable elements of software developed in Phase I. It is performed by the major IPAD contractor mainly as integrator of two development studies subcontracted to aerospace companies on a competitive basis. This phase produces IPAD release capabilities in Systems No. 2 and No. 3 in 51 and 57 months after go-ahead. Estimated development costs include 50 man-months and 6 computer hours for the major IPAD contractor and 296 man-months and 199

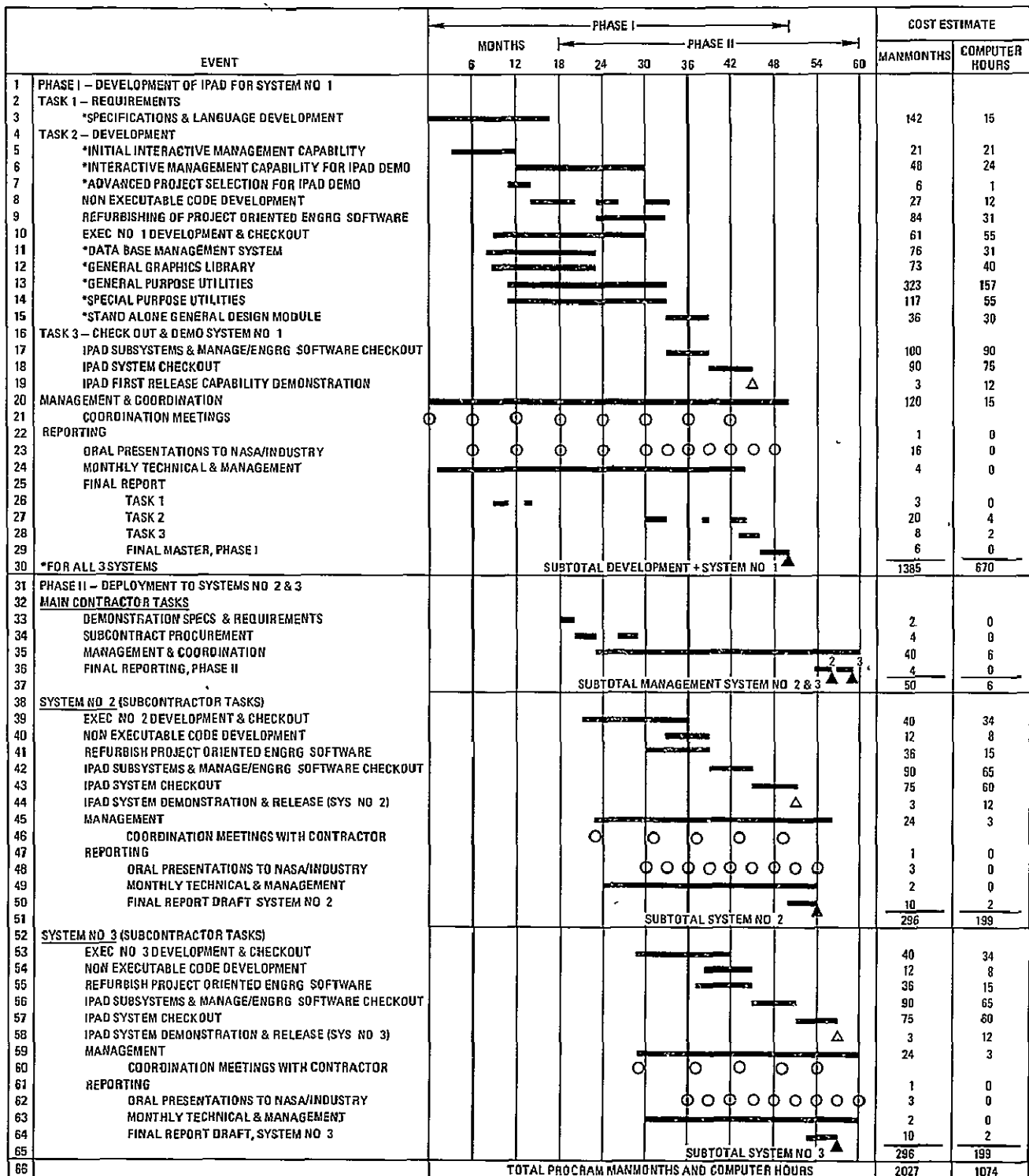


Figure 2-3. Baseline Implementation Plan Schedule and Cost Summary

computer hours for each of the subcontractors, for a total Phase II cost of 642 man-months and 404 computer hours.

The Baseline Implementation Plan is performed during a period of 60 months at a total cost of 2027 man-months and 1074 computer hours.

2.3 Network Planning Technique Used

A variant of the critical path method (CPM) was chosen as the technique to provide schedules, costs, and risk assessment in a single, integrated approach. CPM is a network analysis technique that can provide task dependencies (connectivity), costs, performance times (transport delays), schedules (network summing of the transport delays), etc. on a single diagram. The basic technique was altered slightly for this application. CPM is a block diagram approach; its functional equivalent, PERT, is a nodal diagram approach. Use of a block or nodal diagram technique depends on personal preference.

Figure 2-4 illustrates the basic element of CPM, the task block. Each separately identifiable task is represented as one or more task blocks in the network. Dependencies are reflected by interconnections from the output (right hand) face of a predecessor block to the input (left hand) face of a successor block. All dependencies arriving at an input face must be satisfied before that task can begin. If some portion of that task can begin with the satisfaction of only some of the inputs, and if beginning this subtask will influence the overall schedule/cost/risk, that block must be split into two or more subtask blocks that will reflect this phased work.

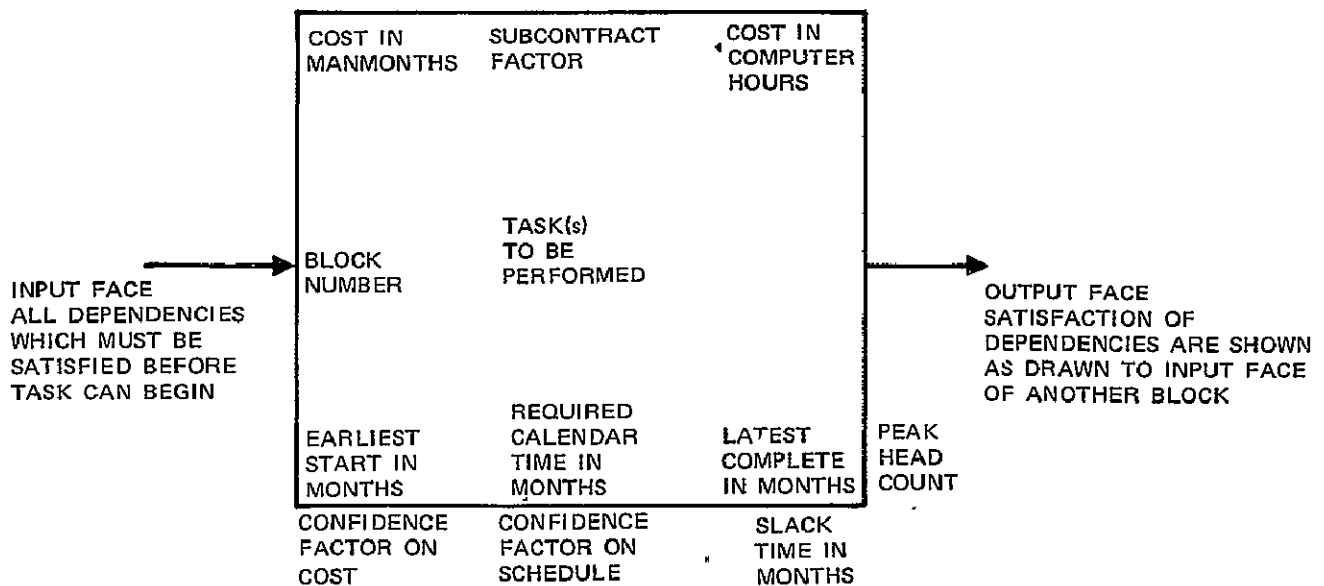


Figure 2-4. Variant of the CPM Task Block

Each block has a unique identifying block number and a descriptive task title. The task title need not be unique (as is sometimes the case with subtask blocks) but should be descriptive of the task to be performed. The purpose of the title is to identify the function of the particular block; the purpose of the block number is to identify the block itself and all its entries. Block entries consist of:

1. Cost in man-months. The estimated number of man-months required to accomplish the task. (The estimates are rounded off to the nearest man-month, which is consistent with the estimate accuracy.)
2. Subcontract factor. The percent of the cost in man-months estimated to be subcontracted.
3. Cost in computer hours. The estimated number of computer hours required to accomplish the task.
4. Earliest start. The earliest month (from the base month, e.g., go-ahead is 1 month zero) for which all of the predecessors blocks tied to the input face are satisfied. (All estimates are rounded off to the nearest month considered to be the minimum task duration).
5. Required calendar time. The number of months required to accomplish the task.
6. Latest complete. The latest month (from the base month) that the task can be completed without delaying a successor block tied to the output face.
7. Peak head count. The estimated peak manpower assignment to the task. This estimate is used to uncover potential manpower-loading problems.
8. Confidence factor on cost. The percentage confidence on the man-month cost figure discussed in Item 1. This constitutes a programmatic risk assessment. A figure of 100 is "complete" confidence that the task can be accomplished for the man-month cost presented, barring unforeseen difficulties that could not be suspected from an in-depth analysis of the task.
9. Confidence factor on schedule. The percentage confidence on the calendar month figure discussed in Item 5. This factor is also programmatic and is treated in the same manner as Item 8.

10. Slack time. The number of months grace in assigning the task. The task may start as late as the earliest start plus the slack time with no effect on schedule/costs (if performed as noted in Items 1, 3 and 5). Slack can be defined as latest complete (Item 6) minus earliest start (Item 4) minus the required calendar time (Item 5), and is a non-negative integer. By definition, any item with no slack is on the "critical path"; i.e., a one-month slippage in this item will result in a one-month slippage of the total schedule.

Several points of clarification regarding the confidence factors are in order. (Note that there is no confidence factor on computer cost, since the net uncertainty in these costs cannot significantly influence the total program costs.) The confidence factor - as assigned - is discussed in the following paragraph.

If the confidence factor as a percent is divided into the cost or schedule estimate for which it purports to be a confidence, the resulting figure is an estimate that the estimator feels confident will not be exceeded, whereas the original figure - although appropriately conservative - assumes good management practices and no significant technical, manpower, or coordination problems. These confidence factors assess programmatic risks only. Technical risks - viz., the risk that a known technical factor will impact the predicted schedule/costs - can best be handled by providing alternative plans for the more critical tasks. Although the alternative plan's programmatic risks can be appropriately assessed, no assessment as to the probability of having to employ this backup plan is offered other than it is of lower priority than the primary plan.

In addition to the basic block of Figure 2-4, the CPM network includes the symbols and callouts shown in Figure 2-5.

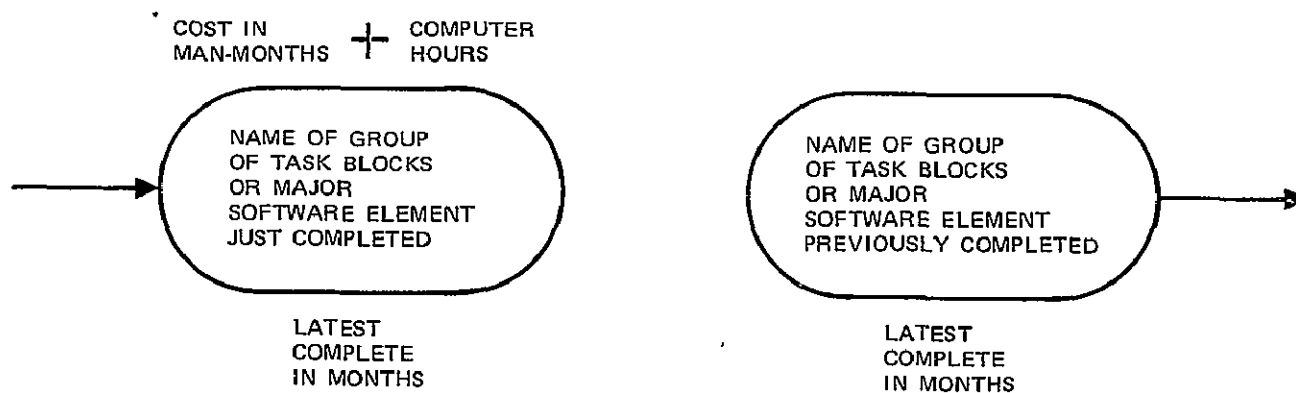


Figure 2-5. CPM Network Symbols

7-10-74 J.A. GIBSON
7-11-74 J.B. GIBSON

The total CPM network is divided into various figures presented in following sections. These figures parallel the Baseline Implementation Plan summarized in Figure 2-3, including its variants and alternatives, and give details associated with each block of the CPM network.

A detailed understanding of the Baseline Implementation Plan and the associated schedule/costs can only be conveyed through a detailed discussion of each task block.

2.4 Phase I - Task 1, Requirements

The objective of this major task is to generate a set of detailed specifications to develop various elements of the computer software required for IPAD, shown in Figure 2-2.

Generation of these specifications has been divided in logical groups, as shown in Figure 2-6. This figure includes activities presumed performed during FY 1974 (in dashed boxes) and activities after contract go-ahead, assumed to be circa July 1975 for the implementation plan discussions in this section. Although costs of the FY 1974 tasks are estimated in the following sections, they are given as reference only in Figure 2-6 and are not included in the cost of the Baseline Implementation Plan, which is assumed to be funded from FY 1975 and on.

The total requirements task is completed 11 months after go-ahead and requires an expenditure of 142 man-months and 15 computer hours. (See Figure 2-3.)

Each group of specifications including its task blocks is discussed in the following subsections.

2.4.1 Operational module (OM) specifications. - These specifications are obtained from the sequence of Blocks 1 through 3, preceded by a nominal effort during FY 1974. The completion date is 11 months from go-ahead at a cost of 14 man-months and 1 computer hour (exclusive of the FY 1974 task). Details on each of these task blocks are discussed in the following paragraphs.

2.4.1.1 FY 1974 task: The objective of this task is to develop a preliminary set of specifications and guidelines for code efficiency, modularity enhancement, and ease of installation of new OMs on the IPAD system. The characteristics of this task are:

1. **Approach.** Many millions of dollars are being invested in the development of new technology yielding coded programs designed around present batch environment. Large programs lacking modularity and interactive interfaces are typically generated in those studies. Recent experiences in the

ORIGINAL PAGE IS
OF POOR QUALITY

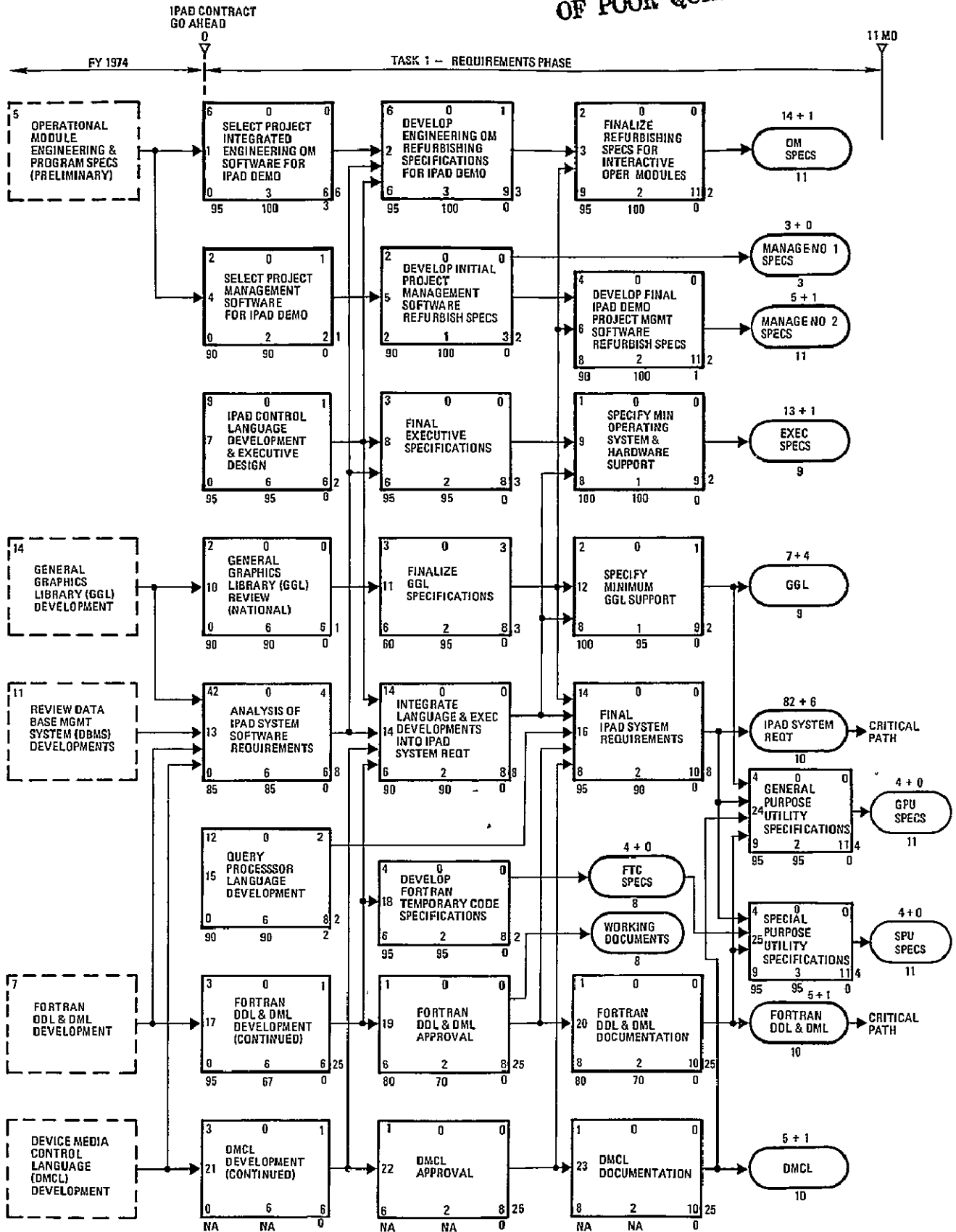


Figure 2-6. IPAD Implementation Plan - Task 1, Requirements

development of IPAD-like integrated/interactive programs point out the need to restructure some existing programs (generated for batch environment) into smaller functional modules both from engineering-user and core-limitation points of view. These specifications will concentrate on new guidelines for the development of disciplinary operational modules (OMs) to exploit the interactive and modular features envisioned for the IPAD design environment, rather than the batch environment. The specifications will be derived using existing programming practice manuals and the experience gained in recently contracted studies to write a preliminary set of specifications and guidelines based on the following considerations:

- a. Each OM designed for an interactive environment to operate in batch mode as well.
 - b. Use of a subset of FORTRAN IV common to various computing systems.
 - c. Modularity in OM structure to facilitate interactive interfaces, multi-level and dynamic overlay structuring, and memory paging.
 - d. Task continuity to provide start and restart capability at convenient control points within OM sequences.
 - e. Portability of source code.
 - f. Addition of interactive interfaces to existing OMs.
 - g. Machine-dependent characteristics.
 - h. Machine/operating system-dependent functions.
 - i. The criteria for selecting re-entrancy of code for multi-user tasks.
 - j. Preliminary planning for future impact of data manipulation and general graphics languages on code enhancements.
 - k. Preliminary planning for the future impact of the data base management system on the structuring of user files.
2. Manpower, Schedule, and Costs. Figures given for this and all other FY 1974 tasks are for reference only and are not included in the Baseline Implementation Plan. The estimated cost of this task is 5 man-months performed during FY 1974.

3. Risk Assessment. There are essentially no risks associated with this task, except the possibility of not being funded during FY 1974.
4. Alternatives. If this task is not funded in FY 1974 it will be performed at the beginning of Block 1, whose task will be slipped back within the slack time available.

2.4.1.2 Block 1: The objective of this task is to select the specific project-oriented computer programs and other engineering OMs to be refurbished and finally used in the IPAD FRC checkout and demonstration. This task has the following features.

1. Approach. A preliminary selection of the type of aerospace vehicle toward which the OMs will be oriented will be made with the concurrence of NASA. Subsequently, from the gamut of automated capability identified in Volumes II and III of this report, appropriate computer programs for the various design/engineering disciplines will be selected. This selection will be made to provide a minimum but meaningful checkout and demonstration base for a project team activity with appropriate multidisciplinary communications and interfaces, and to assemble a multidisciplinary project data base. The set of engineering OMs, though, must be familiar to the engineering team charged with the checkout and demonstration of IPAD. The development of new engineering OMs or the use of existing OMs unfamiliar to the demonstration team should be avoided at this point; there are enough new concepts surrounding the engineering process within IPAD - to which the user must adapt - without introducing new or unfamiliar engineering tools. Further, the success of IPAD depends, fundamentally, on how a project engineering team can exploit the IPAD system using any set of OMs as its members see fit, and this will vary from one aerospace company to another. Special attention will be paid to include general-purpose programs used industry-wide. An excellent example of such a tool is NASTRAN, for which a national panel (user's group) makes improvement recommendations. Such a class would provide an excellent checkout base.
2. Manpower, Schedule, and Costs. A group of six engineers with experience in multidisciplinary design/analysis activities will perform this task for a period of 3 months and a cost of 6 man-months.
3. Risk Assessment. No risk has been assumed, since this task will be performed without significantly influencing costs. It will have a schedule slack time of 3 months.
4. Alternatives. There are no alternatives to the selection of a multidisciplinary set of engineering OMs, except in the extent of the set and the sophistication of its elements.

2.4.1.3 Block 2: The objective of this task is to write the refurbishing specifications for engineering OMs required for a project-oriented activity during checkout and demonstration of the whole IPAD system. This task consists of:

1. Approach. Each of the OMs selected in Block 1 will be analyzed based on the preliminary guidelines developed during FY 1974 to derive a set of specific action items for its refurbishing. Two basic approaches could be used:
 - a. Specify the minimum action items that will permit each OM to run within the IPAD system framework.
 - b. Include improvements in the OMs to enhance engineering user features and/or running efficiency.

Approach b is probably better, but it is not necessary for the major thrust of the implementation plan. These improvements can be added later and since user preference may dictate the extent of the improvements, the user companies may fund this effort individually.

Approach a is essential to IPAD and the extent of the action items will depend on the specific OM. The purpose of Block 2 is to define specific refurbishing specifications for the set of management/engineering OMs.

2. Manpower, Schedule, and Costs. Two computer system programmers and an engineering user will handle this task for a period of 3 months and a cost of 6 man-months and 1 computer hour.

2.4.1.4 Block 3: This task will incorporate the General Graphics Language requirements and finalize the refurbishing specifications for the IPAD-demonstration management/engineering software. The major features of this task are:

1. Approach. Each OM considered in Blocks 1 and 2 will be further investigated to determine its requirements or need for interactive graphics and/or modularity breakdown to enhance or incorporate new interactive interfaces in relation to other OMs.
2. Manpower, Schedule, and Costs. One interactive graphic specialist and an engineering user can supplement the task of Block 2 and write the final set of refurbishing specifications. A period of 2 months is required for a cost of 2 man-months.

2.4.2 Project management software refurbishing specifications. - These specifications are derived from Blocks 4 to 6 and include both an initial capability (MANAGE

No. 1) and a final capability (MANAGE No. 2). The MANAGE No. 1 specifications are completed 3 months from go-ahead at a cost of 3 man-months, while the specifications for MANAGE No. 2 are completed at month 11 and at a cost of 5 man-months and 1 computer hour.

2.4.2.1 Block 4. The objective of this task is to select the project management OMs to be used for IPAD checkout and demonstration. This task consists of:

1. Approach. IPAD project management must perform to increasingly tight schedules attendant with IPAD operation. In an IPAD environment, much data that was previously unavailable for monitoring will suddenly become available and can be monitored with ease. This data collection - which is facilitated by IPAD - can be used to great advantage if suitable tools are made available to the manager. Four categories of OM development have been established to assist the manager in planning, evaluating, analyzing, and deciding. Some of the techniques used in these areas can be found in Reference 1.

The collection of planning OMs shall include as a minimum:

- a. Network modelling programs - used for planning, scheduling, controlling, resource allocation, timeline analysis, cost trading, monitoring, plan integration, etc. Examples of such programs are PERT and CPM. These programs must be able to output conventional timeline diagrams (Gantt charts) and milestone charts.
- b. Forecasting programs - used for planning when supporting data is available and for altering existing plans based on immediate past performance.

The collection might also include:

- c. Resource allocation programs - e.g., job shop models, transportation models, etc.
- d. Layout programs - e.g., equipment or facilities layouts, organization charts, etc.
- e. Inventory control programs,
- f. Simulation programs,

as well as other general planning tools.

The general graphics plotter (GGP) can directly support the assignment of blocks (or nodes), connectivity, and attributes for items a, c, and d. It is assumed for this plan that GGP will be the interface and the OMs can be designed to interface with the network structure GGP builds through DBMS.

The collection of evaluation OMs shall include as a minimum:

- a. Control or trend charts - the charting of variables (or attributes) as a function of a time-related parameter (e.g., design iteration). This is to include the establishing of control limits and trend analysis (forecasting).
- b. Work sampling (sampling theory) - the (usually random) sampling of work accomplished within IPAD, and an evaluation of the adequacy of that work and its performance in accordance with the established plan.
- c. Information Storage and Retrieval (IS&R) - the storage of selected items of information into a logical network structure for quick retrieval via a variety of attributes. For example, cross referencing a collection of reference documents (by name and document number) applicable to a number of subsystems, the storage of regularly sampled information, task status/track of critical path tasks, etc.

The collection might also include:

- d. A management information system (MIS) - a system found generally useful and recommended to a company that may not have an MIS or to an IPAD project whose company's MIS is too comprehensive or awkward to use for the smaller projects.
- e. A special reporting system, e.g., daily reporting by managers of a large IPAD project concerning the status of critical items or recent problems.
- f. Simulation programs - e.g., inventory network analysis, and forecasting.

Of the general purpose utilities (GPUs) STATUM, QP and GGP are likely to play a significant role in the use of the evaluation OMs. Further, a substantial portion of the required code is likely to consist of DDL and TCSSs.

The collection of analysis OMs shall include as a minimum:

- a. Waiting-line models - applied queuing theory consisting principally of discrete system simulators, i.e.:

General purpose system simulator (GPSS) - an IBM development but now available on other computers.

SIMSCRIPT - a development by RAND Corporation, similar to GPSS but not as widely available.

- b. Learning models - the general application of learning curves to general problems.
- c. Assessment models - a generalization of the assessment technique applicable to: risk analysis, value analysis, cost/effectiveness analysis, safety analysis, hazard analysis, threat analysis, maintainability analysis, survivability/vulnerability analysis, reliability analysis, failure modes, effect and criticality analysis (FMECA), and logistic support analysis, as well as other approaches to logic networks (e.g., fault-tree analysis). Each of these techniques employs the same technical approach, although they differ in depth of analysis (compare FMECA with conventional risk analysis) and typically in the input/output form of the data.

The collection might also include:

- d. Linear programming methods.
- e. Gaming - the application of game theory, for example, to collapse many years of decision-making experience into a few hours or explore the possible responses of a competitor. Only the general structure of games should be included together with the provision for formulating the response criteria for the (computerized) pseudo players.

A substantial amount of code already exists in these areas, some of which can be used with little or no modification (e.g., GPSS or SIMSCRIPT) and by providing tutorial TCSSs. The design approach will be to identify an existing, applicable code (where it exists) and modifications required to use this code in the IPAD environment.

It is unlikely that the GPUs, except perhaps for the general graphics plotter (GGP), will have much of a supporting role for this task. This can simplify the required interface with the data bank.

The collection of decision OMs shall include as a minimum:

- a. The formulation of decision criteria via an interactive, auto-tutorial OM. This OM must also guide the user to the proper formulation of the question for which an answer is sought and help decide whether the answer can be obtained through recourse to one of the supplied decision OMs (as well as select that OM).
- b. The decision techniques of decision theory, e.g.:
 - Logic networks (e.g., decision trees).
 - Search techniques.
 - Standard gamble techniques (e.g., Hurwicz's coefficient of optimism, Savage's minimum regret).
 - Heuristic decision rules (assist in their formulation).
 - Delphi technique of soliciting informed judgments.
- c. Standard, widely used models, e.g.:
 - Competitive bidding models.
 - Present value analysis.
 - Bayesian analysis.

It is unlikely that any of the GPUs will be of much direct use to the decision OMs, with the possible exception of QP in obtaining supporting data. The plan assumes that these OMs are "standalone" and principally self sufficient.

2. Manpower, Schedule, and Costs. A project management engineer will perform the task of selecting the appropriate OMs for the IPAD demonstration. This is a full-time task for 2 months at a cost of 2 man-months and 1 computer hour.
3. Risk Assessment. This is a straightforward task having a small risk associated with the appropriate selection of useful and practical OMs of industry-wide use. A 90-percent confidence factor has been assigned both cost and schedule.

4. Alternatives. No alternative is available to this selection task, since a project management capability must be identified for successful operation with IPAD.

2.4.2.2 Block 5: This task will provide the refurbishing specifications for developing an initial interactive management capability (MANAGE No. 1) which can be used both for IPAD demonstration and for managing the IPAD implementation study itself. The major characteristics of this task are:

1. Approach. From the total management capability identified in Block 4, a subset of software will be selected for immediate implementation into an interactive environment using existing equipment and exploiting the computer operating system features available at the contractor's installation. Besides the hardware of the host operating system, the equipment shall include hard-copiers and small interactive terminals supported by existing display software. Timesharing, tasking, and data management capabilities shall be included in the derivation of the refurbishing specifications. Each of the software elements will be analyzed to identify the specific action items that are required for compatibility with existing software and equipment.
2. Manpower, Schedule, and Costs. A computer system analyst with partial support from a management specialist and a graphics system analyst will perform this task in 1 month at a cost of 2 man-months. The confidence factors on cost and schedule are as shown in Block 5.

2.4.2.3 Block 6: This task is similar to that of Block 5, except that it is more involved and must incorporate the results of other task blocks, which define various other features of the complete IPAD system that were not considered for MANAGE No. 1 and that must be included for MANAGE No. 2. This task consists of:

1. Approach. Similar to that of Block 5 but extended to additional management OMs and incorporating IPAD features such as the more sophisticated interactive terminals and the comprehensive data base management system. Refurbishing specifications for the total set of management capability to be used for IPAD demonstrations shall be written.
2. Manpower, Schedule, and Costs. The same team that worked on Block 5 can perform this task in a time space of 2 months and at a cost of 4 man-months.

2.4.3 EXECutive specifications.-- These specifications are the end product of the tasks shown in Blocks 7 to 9 with the connectivity to other task blocks as shown in

Figure 2-6. The EXEC specifications are completed 9 months from go-ahead at a cost of 13 man-months and 1 computer hour.

2.4.3.1 Block 7: The objective of this task is to develop the IPAD control language for use in controlling IPAD's EXECutive and the progress of the user's tasks through the host computing system. Simultaneously, it will extend and finalize the design of the IPAD EXEC. This task has the following characteristics:

1. Approach. EXEC will be intimately associated with the various target computing systems. As such, the design of the ICL must avoid conflict with the host operating system control language (OSCL). Further, it must avoid the general pitfalls uncovered by ANSI committee X3/SPARC/OSCL in their studies of operating system control languages. (See Volume V, Part I, Section 5.1 for an applicable discussion.) This can be accomplished by a careful review of the target computing systems' operating system reference manuals and X3/SPARC/OSCL publications. Conversations with X3/SPARC/OSCL members and the committee itself might additionally be helpful to review the ICL document critically.

The specific functions identified to the EXEC will be finalized simultaneously with the development of ICL. Care must be taken to delegate these functions to the host operating system where the interface has been soundly established. (The objective in designing the EXEC is to minimize its code, where practicable). Operating system modifications desired must be carefully specified and justified.

2. Manpower, Schedule, and Cost. One system programmer/analyst with part-time assistance from another system programmer and several engineering users are required for a period of 6 months and a total cost of 9 man-months. If possible, experience on all three target computing systems should be evidenced by one or both the analysts. The estimates provide for orderly review of documents by both analysts for the first half of the schedule, with follow-through and documentation by the dominant analyst. None of this task is planned for subcontracting due to tight schedule constraints, the need for close coordination and control, and because familiarity of contractor's personnel with this task is required for related tasks further downstream.
3. Risk Assessment. The technical risk relates to obtaining a user-oriented, properly human-engineered language and EXECutive design. This risk is considered low because adequate personnel capability is available and close coordination is to be exercised with potential users. The assignment of two system programmer/analysts to the task with the retention of the

dominant analyst midway through substantially reduces programmatic risks. A 95 percent confidence has been afforded both cost and schedule estimates.

4. Alternatives. Since this is a technically low-risk task, no alternatives have been identified.

2.4.3.2 Block 8: The objective of this task is to compile an EXEC specification suitable for all of the target computing systems. This task consists of:

1. Approach. The EXEC design determined in Block 7 will be specified in sufficient detail so as to be suitable for subcontract purposes. Care must be taken to ensure that all requirements are fully defined and specified. Performance requirements - since these must be demonstrated by the fabricator - present unusual difficulty since system code performance is a function of the job mix in process. Detailed review of the final requirements is required.
2. Manpower, Schedule, and Cost. The same two system programmer/analysts required for Block 7 will accomplish this task with assistance of engineering users and a third system analyst as a reviewer. A period of 2 months is planned for this task at a cost of 3 man-months. No subcontracting of this task is contemplated due to the need for close coordination with Blocks 7 and 9.
3. Risk Assessment. There is little if any programmatic risk associated with this task. A 95 percent confidence is identified to both cost and schedule.
4. Alternatives. No alternatives has been identified, since this is a low-risk task.

2.4.3.3 Block 9: The objectives of this task are to specify for the computer system manufacturers the minimum operating system support to IPAD and to specify for the aerospace companies interested in participating in an IPAD system checkout subcontract - or those intending to use IPAD - the minimum required hardware (computer and terminals) and software required to use IPAD effectively. This task has the following characteristics.

1. Approach. From the EXEC design and the results from the requirements analysis, specify the minimum computing system support for IPAD in quantitative terms. Where different, address each target computing system separately.

2. Manpower, Schedule, and Costs. The same personnel associated with Blocks 7 and 8 can perform this task. A span of 1 month is needed, with a cost of 1 man-month. No subcontract factor is planned for this task.

2.4.4 General graphics library (GGL) specifications. - Blocks 10 to 12 preceded by a development effort during FY 1974 yield a set of GGL specifications 9 months from go-ahead and at a cost of 7 man-months and 4 computer hours.

2.4.4.1 FY 1974 GGL development: The objective of this effort is to establish a preliminary set of specifications for a computer/hardware independent GGL that would enable graphics programs written for GGL to run on any computer system for which the GGL has been supplied. (GGL will be a collection of FORTRAN support packages fielding GGL FORTRAN CALLS and supplying the required function via FORTRAN CALLs to the host system software.) This task consists of:

1. Approach. Considerable national resources are being committed to computer programs written for one manufacturer's computer graphics system (e.g., TEKTRONIK'S PLOT 10, CDC's IGS, UNIVAC's UNIGRASP). Unlike standard FORTRAN programs, these programs cannot be used at other computer installations without extensive rework of the graphics-related code; often related, beneficial - but nonessential - graphics activity is eliminated because it is too costly to provide for, considering the short-term benefits to be derived. More often, these programs are never converted and remain unavailable to users of differing computer installations.

A computer/hardware independent GGL would enable the creation of graphics programs supported by an all FORTRAN GGLs, which would run on any device or computing system that had GGL, within the current limitations of that device or system (e.g., a DVST-type CRT image item can be "selected" - as with a cross hair cursor - but cannot be "picked" since it cannot support a light pen).

The approach will consist of researching existing manufacturer-supplied graphics documentation for (1) function being accomplished, (2) syntax, and (3) lexical structure of their FORTRAN CALLs. From this list, a set of primitives will be developed to supply the framework of GGL. Every effort will be made to provide primitives that (1) do not overlap, (2) can be easily mapped onto existing code, (3) will easily circumvent device or existing computer system software limitations by either automatic substitution of a related capability (e.g., "selecting" replacing "picking") or dropping noncritical functions (e.g., dropping display item "blinking" on a DVST).

A detailed set of specifications will be prepared for these primitives for selected review by key individuals throughout the aerospace and computer-manufacturer community. Selective personal contact will follow for an in-depth critique of the specification with rework to be accomplished as required. Specifications for higher-order building blocks based on the primitives will also be developed and reviewed with the same key individuals to attain concurrence. The final specification will be available for industry-wide review by 1 July 1974.

2. **Manpower, Schedule, and Costs.** The cost of this task is not included in the Baseline Implementation Plan. For reference purposes only, the cost of this task is estimated at 14 man-months and is to be performed during FY 1974.
3. **Risk Assessment.** There are small technical risks associated with developing a set of primitives as well as with the review by key individuals in industry, since highly motivated personnel are available and can be assigned to this task.
4. **Alternatives.** The alternative to this development is to wait for FY 1975 funds. Since GGL development will then be on the critical path, an alternative plan was developed which calls for a GGL committee working for eight months and additional costs (see Section 2 10.1).

2.4.4.2 **Block 10:** This task has the objective of conducting a national, six-month review of the suggested GGL language for IPAD so as to gain and demonstrate a consensus. If a consensus is gained, GGL will become a "de facto" standard, thus eliminating a continuing commitment by NASA to maintain and upgrade GGL. This task will be accomplished as follows:

1. **Approach.** Block 10 is preceded by a coordinated GGL language development assumed to be conducted on FY 1974 funds. The GGL language developed prior to contract go-ahead will be published for industry-wide review/critique. A minimal followup/stimulation will be accorded as a part of the system integration task, including acknowledgement of communications suggesting improvements, etc. All communications will be collected until the review period closes.
2. **Manpower, Schedule, and Costs.** This review is performed by the interested industry at large for a period of 6 months and a cost of 2 man-months. The small costs are associated with coordination and integration efforts by the main IPAD contractor.

3. Risk Assessment. Three technical risks were identified:
 - a. The predecessor task for a coordinated GGL language development is not funded in FY 1974.
 - b. Lack of consensus, with results inconsistent and widely divergent but not necessarily critical of the suggested GGL language.
 - c. Lack of consensus, with results reasonably consistent but critical of the suggested GGL language.
4. Alternatives. If the GGL language development task is not funded on FY 1974 funds (risk item 3.a), the alternative plan calls for forming a GGL committee comprised in such a way as to presume industry consensus if the committee can reach agreement (which it usually can through compromise). This alternative plan is discussed in Section 2.10.

Risk 3.b, where the responses are moderately critical but widely divergent, often results from a superficial response and constitutes the hazard of an unfunded review. The rejection of superficial responses and further review may lead to a successful consensus. If a successful consensus is obtained, the study can proceed with Block 11. If divergent opinion remains, there is little hope that consensus can be attained and the only recourse is to drop the hope for a "de facto" standard and to proceed with an "ad hoc" GGL to meet IPAD needs. The consequence is probably a continuing commitment by NASA to support GGL until consensus is obtained for GGL, or the "ad hoc" GGL gives way to a manufacturer-supported standard and NASA has the GPUs reprogrammed for this standard and slowly phases out the "ad hoc" GGL. In either case, this would occur long after IPAD is operational.

The final risk to be discussed is the lack of consensus through industry-wide agreement in the inappropriateness of the recommended GGL (risk item 3.c). This is possibly the most damaging, since the only viable recourse is to immediately form a GGL committee and proceed to develop a replacement GGL recommendation. Although it is unlikely that the full six months of Block 8 will be required before this becomes evident, some net schedule slippage could occur. The alternative plan discussed in Section 2.10 (excluding the phasing considerations) provides the alternative to this technical risk.

2.4.4.3 Block 11: The objective of this task is to finalize a GGL specification suitable as a reference document supporting Requests for Proposals from potential

subcontractors. This task is planned as follows:

1. Approach. Compile a list of recommended changes to the suggested GGL, make a resolution, and obtain the concurrence of the author of the change. Rework the total specification and carefully review for inconsistencies. Solicit review by key members of industry who have shown considerable interest or insight into the problem. Finalize the specification. To maximize the ability to respond to change, the specification will be entered into a computer with a word-processing capability for quick and accurate text editing.
2. Manpower, Schedule, and Costs. Two graphics programmer/analysts involved with the GGL development (FY 1974) will review the correspondence regarding GGL, make the required resolutions, and modify the text via an interactive computer terminal. Engineering user support will be provided to ensure that human factors associated with usage are properly represented. This task is estimated to last for 2 months at a cost of 3 man-months and 3 computer hours. No subcontracting is contemplated for this task.
3. Risk Assessment. The programmatic risks associated with this block have to do with the size of the task. Due to interactive text editing, the schedule confidence is estimated at 95 percent. However, a large response from industry or major deficiencies in the GGL specs might necessitate a full-time assignment of another graphics analyst. A 60 percent confidence is afforded the man-month cost.
4. Alternatives. The alternatives to this task are directly associated with the alternatives offered under Subsection 2.4.4.2

2.4.4.4 Block 12: The objective of this task is to select a minimum subset of the GGL to support the FRC of IPAD. The features are:

1. Approach. During the assumed FY 1974 development of the GGL language, no restrictions were placed on the capability to be provided by the suggested GGL. Block 12 specifically addresses the question: What is the minimum GGL support necessary for IPAD's FRC? This question can be answered by resolving the requirements of Block 14 with the specifications of Block 11.
2. Manpower, Schedule, and Costs. The principal graphics analyst involved in Block 11 and the requirements analyst associated with Block 14 can effectively complete this task. A span of 1 month is estimated with a cost of 2 man-months and 1 computer hour.

2.4.5 IPAD system requirements.- Blocks 13 to 16, preceded by a FY 1974 effort to review Data Base Management System (DBMS) developments, yield these requirements at month 10, with a cost of 82 man-months and 6 computer hours.

2.4.5.1 Review data base management system developments (FY 1974 task): The objectives of this task are to:

1. Review CODASYL's SCHEMA data description language (DDL), to be published circa July 1973.
2. Review CODASYL documentation on the COBOL data manipulation language (DML) and the COBOL SUBSCHEMA DDL (published in March 1973 and to be available in final form circa March 1974).
3. Investigate viable alternatives to manufacturer-supplied DBMS and to define a fall-back plan for development of an IPAD DBMS capability.

This task consists of:

1. Approach. The following three paragraphs outline the approaches to be followed for each of the three objectives specified.
 - a. The SCHEMA DDL is due to be published in early summer 1973 in the DDL Journal of Development (JOD). It is this document - the first final specification in a series of proposed specifications relating to CODASYL's DBMS - that will provide the basis for IPAD's data management system. Early review and assessment of this document will provide timely response to CODASYL for their consideration and an in-depth evaluation of the capability of the specified SCHEMA DDL to meet IPAD needs.

The contractor will evaluate this document as if it were the basis for IPAD's data base management. Written comments - including recommendations - will be provided to NASA/LRC and CODASYL's DDLC.

- b. Following publication of the CODASYL Data Base Task Group (DBTG) report in April 1971, CODASYL's Programming Language Committee (PLC) formed the Data Base Language Task Group (DBLTG) to develop a detailed set of specifications for COBOL's DML and its SUBSCHEMA DDL. DBLTG's recommended specifications are now available for public comment with the intent of finalizing the specifications for incorporation into the COBOL Journal of Development (JOD) early in 1974. When finalized and published in COBOL's JOD, this document

can provide the basis of CODASYL's contribution to IPAD's Data Base Management. Several factors must be recognized concerning the review of COBOL's DBMS:

Data Base Management capability is on the critical path of IPAD's implementation plan.

The DBMS capability to be provided is independent of the IPAD design approach selected.

This review is timely in that it provides written comments to CODASYL prior to finalization of these proposed specifications.

The contractor will evaluate this document as if it were the basis for IPAD's Data Base Management. Emphasis will be placed on providing a FORTRAN and a data base manipulation capability through COBOL's DML. FORTRAN's lack of RECORD structure will be carefully examined. Written comment - including suggestions for improvement - will be prepared (and reviewed) for submission to NASA/LRC and CODASYL's PLC.

- c. IPAD's orderly development is contingent upon the existence of a DBMS supporting - as a minimum - the COBOL DML and the COBOL SUB-SCHEMA DDL. Only one major hardware manufacturer (UNIVAC) has provided such a system (DMS), in this case based on earlier CODASYL recommendations.

Fortunately, there is at least one data base system based on CODASYL's recommendations that provides a basis for an alternative DBMS to manufacturer-supplied code. The integrated data-base management system (IDMS) developed by B. F. Goodrich (BFG) Chemical Company (Cleveland, Ohio) is based on CODASYL's data base report of April 1971 and is written for IBM 360/370 computers. In point of fact, IBM has a "de facto" DBMS capability through IDMS. IDMS is written in intermediate system language (ISL), which generates basic assembly language for IBM 360/370 systems. IDMS and ISL - both available from BFG for a modest fee - provide an excellent base for extension, not only for recent COBOL DML/DDL developments (by extending IDMS), but also to systems of different manufacturers (e.g., by extending ISL to generate COMPASS code for CDC CYBER systems).

The contractor will investigate IDMS/ISL and other such systems to assess their applicability as a DBMS independent of the computer

manufacturers. Cost and schedule estimates to provide a DBMS capability for UNIVAC, IBM, and CDC will be generated. An evaluation document will be prepared and delivered to NASA/LRC.

2. Manpower, Schedule, and Costs. The cost of this task is not included in the Baseline Implementation Plan. For reference purposes only, the costs of the tasks associated with Approaches a, b, and c are estimated at 3, 2, and 6 man-months respectively. These tasks are assumed to be performed during FY 1974.
3. Risk Assessment. There are no significant risks associated with Approaches a and b. The investigation outlined under Approach c is somewhat more involved and entails one programmatic risk: the task is larger due to technical factors.
4. Alternatives. If this task is not funded in FY 1974, the only alternative is to wait for FY 1975 funds. A delay in the tasks outlined in Approaches a and b will not significantly affect the sequence of Task Blocks 12 to 16 and can be solved by an intensive review early in Block 13. Also evaluations made independently by other companies and agencies may be available early in FY 1975 to reduce the size of the review task. A delay in the task of Approach c, though, will delay the availability of a fall-back plan to provide a Data Base Management capability for IPAD in lieu of the computer-manufacturer supplied code discussed in Section 2.5.4. If this task is completed by the sixth month after go-ahead (i.e. in parallel with Task Blocks 13 and 17), there will be no impact on the program.

2.4.5.2 Block 13: The objective of this task is to define in detail the requirements of each distinct module of IPAD system software and the total system operation, preparatory to system development. This task has three specific predecessors assumed to be accomplished on FY 1974 funding as discussed in the previous subsection. Further, the general development of data-base-related languages by CODASYL and the development of GGL will bear directly on this task. This task has the following features:

1. Approach. This task is associated with industry-wide activities that will yield results of importance to IPAD development. By contract go-ahead, many of the uncertainties present today will have been resolved. In particular, all DBMS requirements supporting COBOL will have been published in specification form. These documents will have been critically reviewed to reassess the ability of these final specifications to meet IPAD's needs (assumed to be a FY 1974 task). Manufacturer commitments - or lack thereof - to provide early implementation of a COBOL DBMS

necessary to support IPAD will be in evidence; a supporting study to investigate viable alternatives to a manufacturer-supplied DBMS will have also been completed (assumed to be a FY 1974 task).

The requirements of each major code module (including the EXEC being conducted in parallel in Block 7) will be examined in detail to ensure that all applicable requirements are specified, can be met, and will support the total system. A Project SCHEMA will be designed and the system operation will undergo a simulated test to help ensure that the system is sufficient and consistent. The final design will evolve out of these requirements lists.

2. Manpower, Schedule, and Costs. A select team of eight computer system and engineering analysts is envisioned for this task for a duration of 6 months, with a cost of 42 man-months and 4 computer hours. No sub-contracting effort is planned for this task.
3. Risk Assessment. Two programmatic risks have been identified:
 - a. The task is larger than anticipated due to technical factors.
 - b. The team composition is such that task integration is difficult.

The first risk can be detected and resolved early by assigning additional support members to the team, thus influencing only cost. The major risk is the second one, which only becomes apparent after the task is underway. The recourse in such matters is close management and coordination but may require recomposing the design requirements team. This often necessitates starting fresh with the new member in evolving the design to the point left off. This risk affects both cost and schedule. An 85-percent confidence was assigned to both schedule and cost to account for both risks.

4. Alternatives. There can be no alternatives to this task, regardless of the technical risks.

2.4.5.3 Block 14: The objective of this task is to integrate additional input into the system design and prepare detailed specifications for each code module of the IPAD system software. This task consists of:

1. Approach. This is an extension of Block 13 incorporating the "most likely" FORTRAN DBMS language interface (Block 17), the status of DMCL developments (Block 21), and the EXEC design (Block 7). This task also

initiates the formal review cycle of the resulting IPAD system design and design requirements.

2. Manpower, Schedule, and Costs. The same design requirements team assigned to Block 13 will continue with this task. In addition, a three-man formal review team will be formed and begin review of the design and design requirements. A performance time of 2 months is planned for this task, at a cost of 14 man-months and no computer hours.

2.4.5.4 Block 15: The objective of this task is to develop the Query Processor Language for use in interrogating, displaying, altering and - in general - managing IPAD's project data base through the Query Processor (QP) general purpose utility. Documentation of the language and its development will also be provided. The characteristics of this task are:

1. Approach. The Query Processor is a general purpose utility that will operate through the computing system supplied DBMS to review data from and effect changes changes to the project's data base or any of its sub-bases. Since DBMS will principally be supporting COBOL in this time frame, QP is envisioned to be a COBOL program which fields requests from the user and converts these into appropriate COBOL DML calls to DBMS. It is these requests - their lexicon and syntax - that comprises QPL. Care must be taken to ensure that the semantics used do not conflict with similar languages and that the semantics form a natural expression of the IPAD user's requests. This can be accomplished by a careful review of the applicable COBOL and DDL Committee documents together with a basic human engineering approach.
2. Manpower, Schedule, and Cost. A single data base analyst with assistance from potential users at selected points is required for a period of 6 months, with a total cost of 12 man-months. This estimate includes the principal analyst, some direct assist by other project personnel, and formal documentation and reviews. None of this task is suitable for subcontract due to close liaison, coordination, and control. Further, skills developed in the progress of this task will be required during the design and checkout of the Query Processor.
3. Risk Assessment. The risks are associated with the quality of the language development such that a properly human-engineered, user-oriented QPL is obtained. Although the technical risk is low, it is not apparent until the data base analyst has had an opportunity to produce some results, whether the results are likely to be of adequate quality or not. If not, the

required corrective action will impact costs and schedule. Doubling up can offset only a portion of the schedule impact. A 90 percent cost and schedule confidence has been identified.

4. Alternatives. Since this is a technically low-risk task, no alternatives have been identified.

2.4.5.5 Block 16: The objective of this task is to integrate the results of Blocks 11, 14, 15, 19, and 22 into a final set of requirements. The task consists of:

1. Approach. This is a direct extension of Block 14 to incorporate the findings of previous connecting blocks.
2. Manpower, Schedule, and Costs. The same team described in Block 14 performs this task for a duration of 2 months and a cost of 14 man-months.

2.4.6 FORTRAN temporary code (FTC), FORTRAN data description language (DDL) and data manipulation language (DML) specifications. - The sequence of a FY 1974 task and Blocks 17 to 20 yield the FTC specifications at month 8, with a cost of 4 man-months (only Block 18 is included here). DDL and DML specifications are prescribed at month 10, with a cost of 5 man-months and 1 computer hour. These costs are only for integration tasks by the major IPAD contractor and do not include the development costs of the FY 1974 nor of Blocks 17, 19, and 20; the Baseline Implementation Plan assumes that these costs will be borne by the companies or agencies sponsoring members of the CODASYL task group proposed herein. If the formation of this task group fails to materialize, an alternative plan with its associated development costs is offered in Section 2.10.

2.4.6.1 FY 1974 task: The objective of this task is to establish a FORTRAN Data Base Language Task Group (FDBLTG) - as a CODASYL task group - for the development of a set of specifications for a FORTRAN DML and a FORTRAN SUBSCHEMA DDL. This task group would publish a set of recommendations specifically for a FORTRAN data base management capability. This task consists of:

1. Approach. The development of a data base management capability solely for IPAD would be a costly and risky venture. Most recent attempts have been formulated following the ground work of CODASYL's Data Base Concepts Task Group (DBCTG), whose recommendations were published in April 1971. The DBCTG - recognizing the role of data base management - made provisions for other languages to use these developments through language-dependent DML and SUBSCHEMA DDL. These language-dependent specifications for COBOL are currently being circulated for public comment prior to being finalized early in 1974. Following publication of these firm specifications, the principal large-scale computer manufacturers are

expected to begin work on their COBOL implementation. Several benefits can be accrued through development of the required language-dependent specification for FORTRAN:

- a. It will probably hasten the commitment of the computer manufacturers to their COBOL implementation (which is to provide the basis for all language implementations).
- b. The initial data base management system (DBMS) in support of COBOL should be a better (e.g., more comprehensive, error-free, and efficient) product.
- c. It will probably hasten the commitment of the computer manufacturers to their FORTRAN implementation - as an extension to their COBOL implementation - thus relieving NASA's IPAD Project Office of the burden (and cost) of maintaining a DBMS code and providing extensions and improvements to it.
- d. The resultant FORTRAN capability should be a much better product for much less cost.

One of the major advantages of the proposed setup is that the IPAD program will benefit both financially and technically from getting the recognized CODASYL organization interested and involved in the development of a fundamental need of IPAD. Preliminary contacts with Mr. Jack Jones - Chairman of CODASYL's Executive Committee - indicates considerable interest within CODASYL regarding CODASYL's assistance in establishing and supervising such a task group. The contractor will foster the establishment of a FDBLTG by:

- a. Discussing organizational and task scheduling details with members of CODASYL's Executive Committee, both collectively and individually.
- b. Generating interest and soliciting participation in such a development within the aerospace community at large and within traditional funding organizations.
- c. Preparing proposals with regard to the mechanisms of forming the task group and some preliminaries on the technical content of a suggested FORTRAN DDL/DML.
- d. Presentations to divulge the benefits to be accrued by such a development.

2. **Manpower, Schedule, and Costs.** The cost of this task is not included in the Baseline Implementation Plan. For reference purposes only, the cost of this task is estimated at 7 man-months.
3. **Risk Assessment.** The programmatic risks of this task are related to uncertainties in the amount of interest that can be generated and how receptive the potential contributing companies and organizations will be to the proposed approach.
4. **Alternatives.** If the FDBLTG is not formed and begins work in 1974, the only recourse will be to revert to one of the alternatives offered in the next subsection. The FDBLTG could be possibly formed under the initiative of other agencies or organizations and if so, the cost of this task will not be incurred.

2.4.6.2 Block 17: The objective of this task is to develop the FORTRAN SUBSCHEMA data description language (DDL) and the FORTRAN data manipulation language (DML) to provide a FORTRAN capability to exploit CODASYL's data base management system and subsequently become an ANSI (or as a minimum, a "de facto") standard. This task has the following characteristics:

1. **Approach.** The task is a continuation of a development presumed started prior to IPAD contractual go-ahead, in FY 1974. It is presumed that CODASYL's Executive Committee will decide to sponsor this development under their Programming Language Committee (PLC). Alternative approaches are presented in the following discussion in case this presumption is not realized.

Preliminary contacts with the chairman of the CODASYL Executive Committee indicate that CODASYL has been concerned over the general lack of programming language development within the scientific community. The American National Standards Institute (ANSI) is responsible for the standardization of programming languages through its ANSC X3 committee dealing with computers and information processing (see Volume IV, Appendix D, Section D.7). The ANSC X3J3 committee is responsible for the maintenance, updating, and clarification, and interpretation of the American National Standard (ANS) FORTRAN just as the ANSC X3J4 committee is responsible for ANS COBOL (see Volume IV, Appendix D, p. D15). CODASYL is recognized internationally as the language development body for COBOL, and ANSI uses the CODASYL work as the base for ANS COBOL; the development of the national and international standard is the combined work of CODASYL for the development phase and the appropriate groups of the European Computer Manufacturers Association,

and ANSI for the standardization phase (see Volume IV, Appendix E, Section E.5). ANSI does not, in itself, develop standards; its only function is to provide the organization through which standards can be approved. (See Volume IV, Appendix D, Section D.7.) Indeed, one of ANSI's prime requirements for consideration of a product has been that it has already been implemented and has found major acceptance in the market place. No formal body (such as CODASYL) is involved with language developments for FORTRAN, which is the basis for CODASYL's concern.

The subject of FORTRAN and its lack of language development has been discussed at CODASYL's last Executive Committee meeting on 29 May 1973 in Washington D.C. together with the possibility of CODASYL's involvement in FORTRAN language development if the scientific community is receptive. CODASYL's basic attitude is "if we can contribute, we are willing to try." There is nothing in CODASYL's charter to preclude this involvement. PLC's charter talks only of programming language development in general and led to the recommended specifications of the language-independent SCHEMA DDL and DBMS by their Data Base Task Group in 1971. It is uncertain what technical approach the new CODASYL task group will adopt. The following possibilities exist.

- a. Use the COBOL-SUBSCHEMA DDL and the COBOL DML as they are, intermixed with the FORTRAN source, or use their syntax, but organized as FORTRAN CALLs.
- b. Use the COBOL SUBSCHEMA DDL, but develop a special DML for FORTRAN (either CALL statements or new FORTRAN statements).
- c. Develop a special DML and SUBSCHEMA DDL for FORTRAN, patterned after COBOL.
- d. Develop a special DML, SUBSCHEMA DDL and SCHEMA DDL for FORTRAN, patterned after COBOL and the existing SCHEMA DDL.

It is unlikely that the new task group will alter the concepts embodied in the existing DML and SCHEMA/SUBSCHEMA DDLs. (That is, the developed FORTRAN language will probably map directly to its counterpart in COBOL or the SCHEMA, with only the lexical and syntactical structure changed. This does not preclude, however, that the precedence relationships cannot be partially delegated to the FORTRAN object-code supporting subroutines, some of which would contain the user work area.)

2. **Manpower, Schedule, and Costs.** Presuming that the task group is formed, it will organize as a CODASYL task group funded indirectly through organizations sponsoring membership on the task group. These organizations need not be members of CODASYL. Typically, the task group would be composed of no more than 25 members, with adequate user and implementer representation across the FORTRAN community. These members would all be data management analysts.

According to past schedules, approximately two years were required for a CODASYL Task Group (or working standing committee) to draft and agree upon a language specification once the basic concepts had been agreed on. For example, CODASYL's Data Base Task Group published a preliminary set of recommendations for a Data Description Language for the SCHEMA in 1969; the SCHEMA DDL was revised and presented in a set of recommended specifications in April 1971. CODASYL's Data Description Language Committee revised the SCHEMA DDL and approved the final specifications for inclusion in the DDL Journal of Development (JOD) in April 1973. Meanwhile, CODASYL's Programming Language Committee had furthered work on the COBOL DML and SUBSCHEMA DDL (contained in the April 1971 report) and published its recommendations for public comment in February 1973. Comments were solicited and a final set of specifications will be approved for inclusion in the COBOL JOD early next year (circa February 1974). Hence, the normal CODASYL developmental cycle can be thought of as a minimum of two years.

Several factors could reduce this developmental schedule for FORTRAN. The Data Description Language Committee was as involved with concepts as it was with language specification, and the resulting SCHEMA DDL had to satisfy all language applications. However, the development of the FORTRAN SUBSCHEMA DDL can be best likened to that for COBOL, and the COBOL developments can provide a sound basis for comparison. During this development the Data Base Language Task Group of the Programming Language Committee did not have the final SCHEMA DDL specifications to work from. (These were not available until about May 1973.) Further, this Task Group was pioneering the development of a DML and a SUBSCHEMA DDL. Such is not the case with the FORTRAN developments, since they can be patterned after COBOL. Indeed, several implementers of DBMS-like systems for FORTRAN have adopted the COBOL DML syntax for their FORTRAN DBMS CALLs.

It is uncertain exactly when the task group might be formed and when it might begin its task. Block 17 presumes that the task group has been working for nine months; the indicated six-month period for Block 17 is

considered sufficient to finalize a set of specifications. A significant amount of developmental work will be accomplished with no direct costs identified to the IPAD implementation plan. That is, presuming that the CODASYL task group is formed, the costs associated with language development are not identified to the IPAD project in this Baseline Implementation Plan. The costs associated with alternative approaches are discussed in Section 2.10. The costs shown in Block 17 pertain to integration tasks performed by the major IPAD contractor.

3. Risk Assessment. The programmatic risk associated with the cost of Block 17 is minimal. The schedule, however, is a different problem. Presuming that the preceding development work is completed, the confidence that the language development will have progressed to the point required to support Block 17 in the subsequent 6 months is 67 percent. (It is highly unlikely that this task should require more than nine months.)
4. Alternatives. Two alternatives were identified in lieu of a CODASYL-sponsored task group:
 - a. Undertake the task as a contractor's task to be completed by several individuals.
 - b. Form an "ad hoc" task group to develop a recommended FORTRAN DML and SUBSCHEMA DDL.

The first alternative was dismissed as not meeting the objective of becoming - as a minimum - a "de facto" standard. It is considered highly unlikely that a small group of individuals - essentially from the same company background - could generate so detailed a specification and obtain the consensus required of a national standard. The second alternative calls for forming a seven-man task group on contract funds. (This task group might even be indirectly sponsored by CODASYL.) This alternative plan is discussed in detail in subsection 2.10.2.

Several options are available in case CODASYL structures the 25-member task group but the FORTRAN community does not volunteer qualified members and the task group fails to be formed. These options are:

- a. Fund task group members via small NASA contracts to the companies of qualified potential participants.
- b. Fund members (as in Item a) with a combination of small contracts from a variety of funding organizations (e.g., NASA, DOD).

- c. Revert to the ad-hoc seven-man task group approach discussed above and in Section 2.10 using funding as in a or b.

It is uncertain which approach might be selected by NASA, and it would depend on the degree of response from FORTRAN community and the interest developed within the funding organizations. Since this is all highly problematic, only the seven-man task group alternative was explored. (See Section 2.10.)

2.4.6.3 Block 18: The objective of this task is to bring about a set of FORTRAN temporary code specifications for satisfying the FORTRAN DML and FORTRAN SUBSCHEMA DDL through COBOL, as a temporary expedient. This task consists of:

1. Approach. As the implementation plan took form, it was recognized that only a little over a year remained between the firming up of the FORTRAN DDL & DML (Block 20 terminates at month 10) and IPAD's need for DBMS support to FORTRAN. (Figure 2-10 in Section 2.5.4 shows a constraint of not later than 23 months.) This is too little time to secure the manufacturers' backing of FORTRAN support from DBMS and for them to fabricate the required code; that is:
 - a. An extension to their FORTRAN compiler(s) to process the DML statements if these take the form of a language extension rather than FORTRAN CALL statements.
 - b. An extension to their data base management subsystem to honor the FORTRAN DML (if required).
 - c. An extension to their DDL compiler(s) to process the FORTRAN SUBSCHEMA (and perhaps also the SCHEMA), which is presumed written in FORTRAN DDL.

Since this was the case - and to avoid the schedule uncertainty attendant to this approach - the alternative plan of providing FORTRAN DBMS support via COBOL was adopted as the baseline plan. This, in turn, necessitated a prejudgement of the FORTRAN DML/DDL developments (Blocks 17, 19, and 20).

The capabilities of DML will probably be provided through standard FORTRAN CALL statements. CALL statements have the distinct advantage that only the syntax of the variables being passed through the argument lists are specified; the programmer is free to select names for the argument variables. This leaves only the subroutine name and argument

syntax, which must be specified to tailor the COBOL DML to FORTRAN applications. Further - from the implementer's viewpoint - this is the simplest method of extending the host language, since it does not require a compiler modification. Two pioneering FORTRAN implementations of the DBTG recommendations (References 3 and 4) have already taken this approach. Indeed, this technique is so widely used in providing software support to the FORTRAN language that it is almost inconceivable that the CODASYL FORTRAN DML/DDI committee would recommend any other approach.

The approach to an acceptable SUBSCHEMA DDL is not so obvious. A certain amount of data description capability currently exists within the FORTRAN language and the committee will probably investigate using this in the SUBSCHEMA DDL. However, this capability corresponds to a part of the Data Sub-Entry of the Record Entry. Essentially, the rest of the DDL concepts are new to FORTRAN, so the basic task of the committee is to present these concepts in a FORTRAN-like language. The aforementioned pioneering implementations of the DBTG recommendations took the expedient of incorporating DDL as per the DBTG report. However, a committee concerned more with language development than with implementation will surely attempt to:

- a. Condense and abbreviate, thereby specifying codes where the DBTG used COBOL-like "clauses".
- b. Provide default conventions to replace specifications wherever possible, in keeping with standard FORTRAN. (In standard FORTRAN for example, all variable names beginning with conventional characters are assumed to be TYPE IS FIXED with a default precision. The programmer can specify all exceptions to the rule in a single statement or make many statements of exception.)
- c. Provide specifications through ordered lists or tabulations.

What can be said for the SUBSCHEMA is equally applicable to the SCHEMA, where - in many FORTRAN applications - the programmer himself will probably act as a local Data Base Administrator as well. The desire for a more compact SCHEMA DDL supporting the FORTRAN SUBSCHEMA will probably result in an investigation into a companion recommendation for a FORTRAN SCHEMA DDL.

In summary, the best estimate is that the FORTRAN developments will probably result in FORTRAN SUBSCHEMA DDL, SCHEMA DDL, and DML

via subroutine CALLs, and the approach to Block 18 is to formulate the required specifications based on the in-process work of the FORTRAN DML/DDI Task Group that performed the task in Block 17. These specifications will be updated during subcontract negotiations (reference Figure 2-10) to incorporate the changes that might have resulted from Blocks 19 and 20.

2. Manpower, Schedule, and Costs. A data base analyst and a systems programmer/analyst will team to draft these specifications during a period of 2 months, with a cost of 4 man-months.

2.4.6.4 Block 19: This is a continuation of Block 17. The objective of this task is to review and approve the final specifications for submittal to the parent group within CODASYL (assumed to be the Programming Language Committee) and subsequently to the CODASYL's Executive Committee. The task consists of:

1. Approach. This is a standard CODASYL task group approach. Having preliminarily agreed upon a set of specifications (Block 17), the task group adjourns and each member subsequently receives a computer listing containing the detailed specification. Each member spends the available time prior to the next meeting reviewing the specification in detail and noting suggested changes. They then meet, review the specification page-by-page, noting final corrections to be incorporated and, if in consensus, approve it for publication.
2. Manpower, Schedule, and Costs. The same CODASYL task group of Block 17 performs this task. Due to the urgency of the specification to meet IPAD's needs, it is presumed that the task group will respond without undue delay. The longest potential delay is involved with getting the computer listing containing the specification into the hands of the task group members for review. It is intended to offer the services of the IPAD contractor (using system integration funds) to record (using an interactive text editor with word-processing capability, including upper and lower case) and make available to each member a complete listing of the specification. In less than one month, each member will have a complete review copy. An additional month is presumed to be an adequate review period. A total of two months is presumed adequate to bring the specification to a vote. The costs shown in Block 19 are only for integration tasks by the IPAD contractor.
3. Risk Assessment. The programmatic risks are: slippage of the listings, request of members for more than one month review, and scheduling problems for the final review meeting. The technical risk associated with this

task is failure to achieve consensus. A 70 percent schedule confidence was assessed.

4. Alternatives. Failure to achieve consensus generally occurs because the majority of the voting members feel that additional work is needed: i.e., that the specification is inadequate as it stands. Since Blocks 17 and 19 are on the critical path, there is no recourse except to be patient, aid in the additional work deemed necessary by the task group, and strive for an early resolution and favorable vote. Several months could be added to the IPAD baseline schedule if this occurs.

A much less likely reason to reach a consensus is that the majority of the voting members feel it is too early to specify a language for FORTRAN (e.g., because it may stifle creativity) or that the total approach is wrong. Usually these decisions are formed much earlier and the result is redirecting or disbanding the task group. Redirection might result in schedule slippage. Disbandment will likely occur before contract go-ahead and a fall-back position is provided by the alternative plan in Section 2.10.

2.4.6.5 Working documents from Block 19: These documents are needed immediately to start the development of the Data Base Management System, without having to wait for the final documents generated under Block 20.

2.4.6.6 Block 20: This is a continuation of Block 19. The objective of this task is to document the developed specification as an official CODASYL document, eventually intended for adoption by ANSI as a FORTRAN standard. This task consists of:

1. Approach. CODASYL - through its Executive Committee - publishes three categories of reports as official documents:
 - a. Ideas report - to get distribution to and feedback from the community at large on a collection of ideas. Even minority reports can be published in this category, a consensus not being a requirement. The intent is to inform the community that these ideas have been advanced, CODASYL is considering them, and CODASYL would appreciate all comments. This is the easiest type of report to get published. An example is the 1969 DBTG report.
 - b. Proposal report - to get distribution to and feedback from the community at large on a proposed specification prior to official adoption of a specification by CODASYL. This report must have a consensus for it is expected to be the basis for the specification. An example is the February 1973 DBLTG COBOL report.

- c. Journal report - publication in the Journal of Development for that language is official adoption by CODASYL of those recommended specifications. In the case of COBOL, ANSI adoption of the standard usually follows in due course. This is the most difficult type of report to get published and generally must be preceded by a proposal report.

The approach of Block 20 presumes the publication of a proposal report. Not shown on Figure 2-6 is a review period, followed by a rework of the intended specification for publication in the JOD. This is not shown because IPAD development cannot await a final specification and must proceed with the proposed specification on a risk basis. What is hoped for is a high-quality proposal report that will undergo minimal change when upgraded to journal report. This will minimize NASA's long-range commitment to IPAD.

2. Manpower, Schedule, and Costs. The same task group of Blocks 17 and 19 performs this task. To avoid unnecessary delays, it is intended to continue the services of the IPAD contractor (using system integration funds) to update the final report and make these available to the task group, PLC, and CODASYL's Executive committee.

The critical schedule is associated with technical review by PLC. It is assumed that this can be accomplished within two months. Final approval by the Executive Committee - although not guaranteed - will generally follow approval by PLC. It is within the intent of the baseline implementation plan that following approval by PLC, and prior to approval by the Executive Committee (if this be the case), copies of the as yet unapproved document will be made available to potential subcontractors of IPAD tasks. CODASYL, of course, assumes no responsibility for subsequent changes in the document prior to publication. There is little reason to suspect that concurrence to this approach will be denied by CODASYL. In this way, final publication of the document can be temporarily circumvented - if required - to meet IPAD's schedule. The cost shown in Block 20 is associated only with integration tasks by the IPAD contractor.

3. Risk Assessment. Circumventing final publication of the document, the programmatic risks are associated with slippage of the listings, delay in the balloting, request from PLC for more than two months for review, and scheduling problems for the PLC review meeting. Two technical risks have been identified:

- a. A rejection of the report in final balloting.

- b. A request for additional work by PLC prior to their approval as a proposal report.

The first risk is highly improbable, since it necessitates that enough members of the task group withdrew their commitment between final approval and official balloting to overcome the majority. The second risk is real, especially with a newly formed task group. If this were the case, it is unlikely that the additional work and review could be accomplished without major impact on IPAD's schedule. A confidence of 70 percent has been assigned to the schedule. The cost shown in Block 20 pertains to integration tasks by the IPAD contractor.

4. Alternatives. The technical risks lead to two recourses:

- a. Press for a degrading of the document from Proposal to Idea report and subsequent quick approval.
- b. Accept the schedule slippage.

Which recourse is appropriate can only be judged at the time and depends on the nature of PLC's critique and on the prognosis of an early resolution. In summary, the realistic alternatives in lieu of PLC approval is to elect a schedule delay or press for an Ideas report. This uncertainty can only be resolved at that time.

2.4.7 Device/media control language (DMCL). - The sequence of a FY 1974 effort followed by Task Blocks 21 to 23 provide the documentation for the DMCL 10 months from go-ahead, with a cost of 5 man-months and 1 computer hour. These costs are for integration tasks by the major IPAD contractor only and do not include development cost for the FY 1974 task nor for Blocks 21, 22, and 23; the Baseline Implementation Plan assumes that these costs will be borne by the companies or agencies sponsoring membership in the CODASYL Data Description Language Committee (DDLC). This is a noncritical development for IPAD, since sufficient capability exists within the operating system control language (OSCL) of the major computing systems to support an IPAD FRC. The DMCL capability can be incorporated into IPAD later. Nevertheless, details of the effort required for its development are given in following subsections.

2.4.7.1 FY 1974 task and Block 21: The objective of these tasks is to develop a language-independent, computing-system-independent DMCL for allocating data-base information handled by DBMS to specific devices (e.g., drum, tape) or media (e.g. listings, cards). The characteristics of these tasks are:

1. Approach. At their April 1973 meeting, the DDLIC approved for publication (in the first issue of the DDL JOD) the specifications for a language-independent DDL. The prime subject for discussion at their next meeting will be the next task to be undertaken by DDLIC. This will probably be development of DMCL.

Device/media control is the missing link in the DBMS capability. Most operating systems have limited DMCL as part of their OSCL, but this is insufficient to fully support DBMS operation. DMCL must come about; since it is in DDLIC's charter, DDLIC must ensure that it is computing-system independent and meets the needs of DBMS. DMCL will probably be undertaken by DDLIC by late 1973. It is questionable, however, that a recommended specification (JOD report) will result within a two-year development cycle due to the complexity of device/media control, the degree to which this has already been addressed and is in evidence within the various OSCLs, the required interface with DBMS, and the conceptual development that must accompany the language design. This can be summarized as follows, and several conclusions can be drawn immediately:

- a. DMCL must be developed in support of DBMS. This will probably be undertaken by DDLIC in 1973 but is expected to be an involved task.
- b. This task is of sufficient complexity that a special IPAD effort would not be worth the high cost, since the results of a special effort would be relatively short lived, being subsequently replaced by DMCL, and sufficient DMCL exists within the OSCL of the target computing systems to support IPAD's FRC.
- c. The supporting code probably cannot be completed in time to meet IPAD schedules; in particular, the manufacturers will not have had an opportunity to generate much (if any) code in support of DMCL by IPAD's PRC.
- d. When DMCL becomes available on the host operating systems, this added capability can be used effectively in support of an IPAD project.

The latter consideration gives rise to task Blocks 21, 22, and 23; in this regard, they are reference blocks of an influential but noncritical development.

The approach is to monitor the progress of DMCL developments and incorporate these considerations into the final design of IPAD. For this purpose, the development plan assumed is patterned after that for the

FORTTRAN DDL and DML (Blocks 17, 19, and 20) and assumes that a proposal document results.

2. **Manpower, Schedule, and Costs.** This task is presumed to be performed by the Data Description Language Committee, which is a 25-member CODASYL committee that could function exactly in the manner discussed in Section 2.4.6.
3. **Risk Assessment.** Since the intent is only to monitor the progress of DMCL developments and incorporate these considerations into the final IPAD design, the development of DMCL is noncritical to IPAD's FRC. As such, no programmatic risks should be assessed.
4. **Alternatives.** This is a noncritical development. The only technical risk remaining is that CODASYL (DDLC in particular) elects not to undertake DMCL development at this time. This would be known by contract go-ahead, and there will be nothing to monitor. Two alternatives can be identified:
 - a. Do nothing, presuming that CODASYL must undertake these developments sooner or later, and they can then be applied to IPAD.
 - b. Initiate a special IPAD effort to fill the void.

As previously discussed, it is doubtful that this effort would be worth the high cost, even if CODASYL delayed the initiation of DMCL for two full years. If - at that time - it was judged unlikely that CODASYL would be undertaking DMCL in the near future, a special IPAD effort might be justified. This decision could only be made at that time and is considered problematical; it is highly unlikely that CODASYL would essentially abdicate this responsibility. The alternatives, then, reduce to just one. If for some reason CODASYL delays development of DMCL, do nothing.

2.4.7.2 **Block 22:** This block is patterned after Block 19, but in reference to the DMCL.

2.4.7.3 **Block 23:** This task is similar to that of Block 20.

2.4.8 General purpose utility (GPU) specifications. - The objective of Block 24 is to write the detailed specifications for each of the IPAD GPUs. This task consists of:

1. **Approach.** Integrate the results of Blocks 12, 16, 20, and 23 into a final set of requirements for the five GPUs; that is STATUM, OPTUM, Query

Processor, General Graphics Plotter, and General Design Module.

2. Manpower, Schedule, and Costs. A team of four people including an engineering user, a data base analyst, a graphics programmer, and a requirements analyst will perform this task for a period of 2 months, with a cost of 4 man-months.

2.4.9 Special purpose utility (SPU) specifications. - The objective of Block 25 is to write the detailed specifications for each of the IPAD SPUs. The features of this task are:

1. Approach. Integrate the results of Blocks 16, 18 and 20 into a final set of requirements for the three SPUs; that is, the DML Insertion Preprocessor, the SUBSCHEMA Assembler, and the SCHEMA Assembler. Since details of these SPUs are heavily dependent on the final design of the SCHEMA DDL and the FORTRAN DML and FORTRAN SUBSCHEMA DDL, finalization of the SPUs must await the FORTRAN DML/DDI documentation (Block 20) as well as the final requirements of the IPAD system (Block 16). In addition, constraints imposed by the FORTRAN temporary code (FTC) in Block 18 will influence the design of the DML Insertion Preprocessor.
2. Manpower, Schedule, and Costs. A team of four people, including a data base analyst, a systems programmer, a requirements analyst, and an engineering user, will complete this task in 3 months, with a cost of 4 man-months.

2.5 Phase I - Task 2, Development

The objective of this major task is to code or refurbish each element of software associated with IPAD. (See Figure 2-2.) This development task is completed 33 months from go-ahead at an expenditure of 872 man-months and 456 computer hours

A discussion and implementation plan for each software element are given in the following subsections.

2.5.1 Project management capability. - Figure 2-7 shows the plan for developing this capability in two subsets: MANAGE No. 1 and MANAGE No. 2.

MANAGE No. 1 is an initial Project Management capability to be used both for IPAD demonstration and for managing the IPAD implementation contract itself. It is completed 12 months from go-ahead, with a cost of 21 man-months and 21 computer hours.

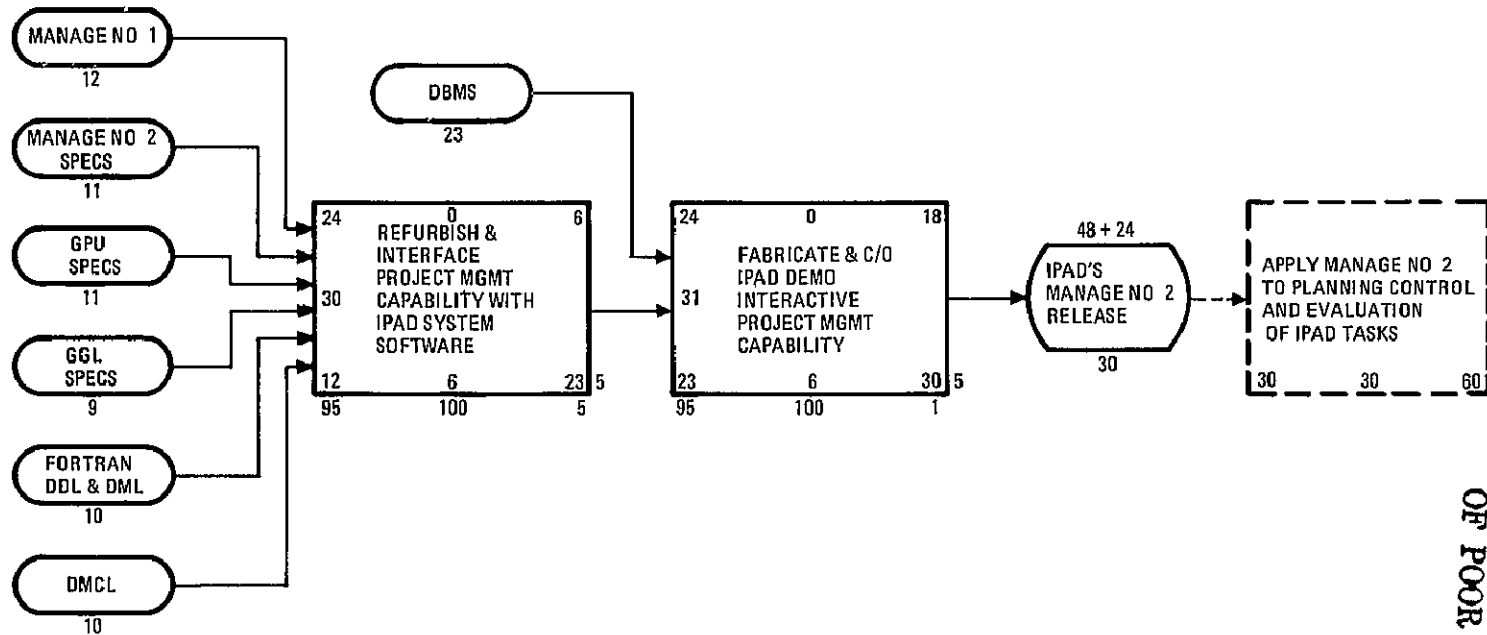
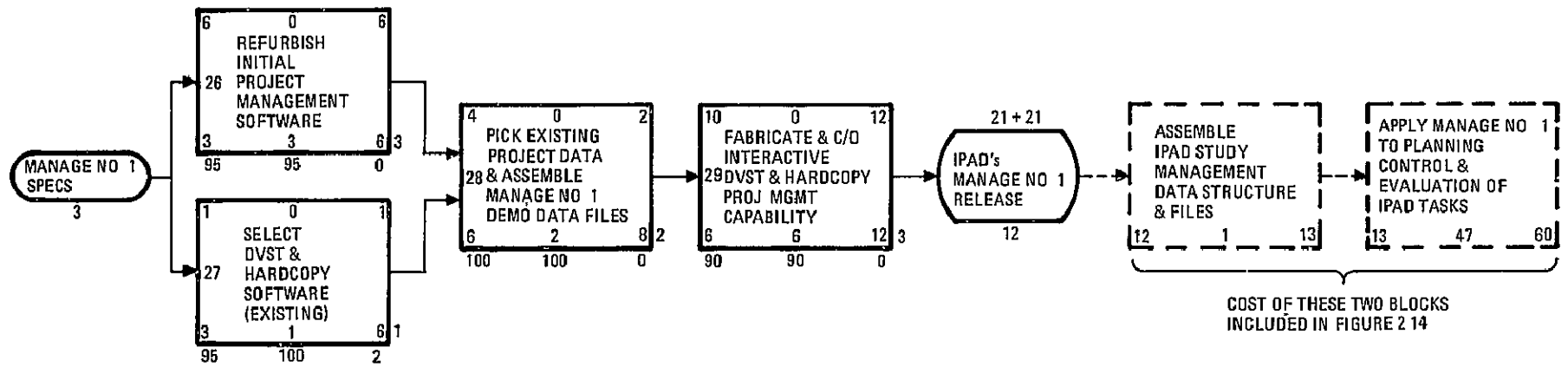


Figure 2-7. Development Phase Project Management Capability

ORIGINAL PAGE IS
OF POOR QUALITY

MANAGE No. 2 is the final Project Management capability that will exploit all features of the IPAD System and will be used for IPAD demonstration. This capability is developed by 30 months after go-ahead, with a cost of 48 man-months and 24 computer hours.

2.5.1.1 Block 26: The objective of this task is to refurbish the software selected for MANAGE No. 1. The task consists of:

1. Approach. The specifications developed in Block 5 will be used in refurbishing existing capability in the areas of control and trend charts, work sampling, information storage and retrieval, waiting-line models, cost-effective analysis, network modeling, simulation, and forecasting.
2. Manpower, Schedule, and Costs. A team of a graphics programmer, a systems programmer and a project management engineering user will perform this task for a period of 3 months, with a cost of 6 man-months and 6 computer hours.

2.5.1.2 Block 27: The objective of this task is to select the specific devices to be used in conjunction with Block 26. This task consists of:

1. Approach. The equipment available at the IPAD contractor's computing facilities will be evaluated from the human engineering point of view and the results of Block 5 to select the specific devices on which the MANAGE No. 1 capability will be exercised. Interactive graphics terminals of the direct view storage tube (DVST) and refreshed types will be compared on cost-effective and ease-of-deployment bases, solely for the purpose of mounting the MANAGE No. 1 capability and making it available to potential users. Hard-copying equipment will be also evaluated and selected on a similar basis.
2. Manpower, Schedule, and Costs. A graphics programmer with close assistance and direction from a facilities supervisor and the project management user of Block 26 will perform this task. The duration and cost of this task is estimated at 1 month and 1 man-month plus 1 computer hour respectively.

2.5.1.3 Block 28: The purpose of this task is to pick a set of management and engineering data from an existing document and assemble the data files that will be required to check out and demonstrate MANAGE No. 1. The task will be accomplished as follows:

1. Approach. Existing documentation from recent contractor or NASA studies

on an aerospace-vehicle project will be reviewed to segregate appropriate project management data on sequentially releasable blocks to simulate a time-phased evolution of the project. The reason for selecting existing project documentation is to obtain a complete set of data at a very nominal cost.

2. Manpower, Schedule, and Costs. A team consisting of a project management engineer and a computer system programmer will complete this task in 2 months at a cost of 4 man-months and 2 computer hours.

2.5.1.4 Block 29: The objective of this task is to fabricate and check out the interactive MANAGE No. 1 capability. This task consists of:

1. Approach. The sequence of Task Blocks 26 to 28 provides all information required to modify, expand, and adapt existing project management software to mount it on existing interactive devices and provide a user-oriented capability. Each of the software modules will be refurbished on a stand-alone basis first, followed by the total checkout/demonstration using the sequential project data blocks selected in Task Block 28.
2. Manpower, Schedule, and Cost. A team of a project management user, a graphics programmer, and a computer systems programmer will perform this task in 6 months for a cost of 10 man-months and 12 computer hours.

2.5.1.5 Application of MANAGE No. 1 to planning and control of IPAD contract tasks: Once the management capability is properly checked out and demonstrated, it will be exercised on the remaining tasks of the IPAD contract under which it was developed. The characteristics of this task are:

1. Approach. In the first month after completion, the IPAD study management-data structure and files will be assembled. These files will include the program history of the previous year as well as all planning charts and projections developed to that time.

The subsequent period of 47 months will be monitored by this interactive capability. MANAGE No. 1 will be absorbed to become part of MANAGE No. 2, such that duplication of effort will be minimized.

2. Manpower, Schedule, and Costs. The IPAD contract manager will be the major user and field tester of MANAGE No. 1. The man-hours required to apply this capability to the IPAD contract are borne by the management tasks described in Section 2.7 and are expected to be fewer than required by conventional methods, resulting in net cost savings for this portion of the study.

2.5.1.6 Block 30: This is the first task block leading to the development of MANAGE No. 2, the final project management capability for IPAD demonstration. The objective of this task block is to interface all the project management software with the IPAD System software specifications, and refurbish the former accordingly. This task consists of:

1. Approach. All management software selected in Block 4 and its refurbishing specifications developed in Block 6 will be interfaced with the specifications for the general purpose utilities (GPU), the general graphics library (GGL), the FORTRAN data description and data manipulation languages (DDL & DML), and the device/media control language. The management software elements will then be modified, expanded, and adapted to exploit all features envisioned for IPAD that emphasize the user and cost savings.
2. Manpower, Schedule, and Costs. A team of five members consisting of a project management engineer, a data-base analyst, a graphics programmer and two computer programmers will perform this task in 6 months at a cost of 24 man-months and 6 computer hours.

2.5.1.7 Block 31: The objective of this task is to fabricate and check out all interactive management capability refurbished for IPAD demonstration, including the interfacing with the data base management system (DBMS) completed at month 23. This task will be accomplished as follows:

1. Approach. Each software module will first be checked out on a stand-alone basis then integrated through the DBMS capability. All types of interactive devices, including large-screen display equipment, will be exploited to provide the visibility required from a comprehensive project management task.
2. Manpower, Schedule, and Costs. The same team as for Block 30 will perform this task in 6 months, with a cost of 24 man-months and 18 computer hours.

2.5.1.8 Use of MANAGE No. 2 in IPAD contract study: Since this capability is available 30 months from go-ahead (mid-point in the proposed Baseline Implementation Plan), it will be used for the remainder of the contract. The new capability, which is not available in MANAGE No. 1, will afford a more cost-effective management function and is expected to provide additional cost savings.

2.5.2 Project-oriented management/engineering software and nonexecutable Code No. 1. - The objective of this sequence of tasks is to assemble a set of automated capability typical of that required for an aerospace-vehicle project-oriented activity. This

capability is needed for a comprehensive checkout and demonstration of the IPAD system. The tasks, illustrated in Figure 2-8, include selection of a model aerospace-vehicle project, refurbishing of software for various disciplinary groups, development of non-executable code, and integration and checkout of management and engineering capability within the IPAD system. This effort is completed 33 months after go-ahead, with a cost of 90 man-months plus 32 computer hours for the project-oriented software, and 27 man-months and 12 computer hours for the non-executable code. Details for each task block are presented in following subsections.

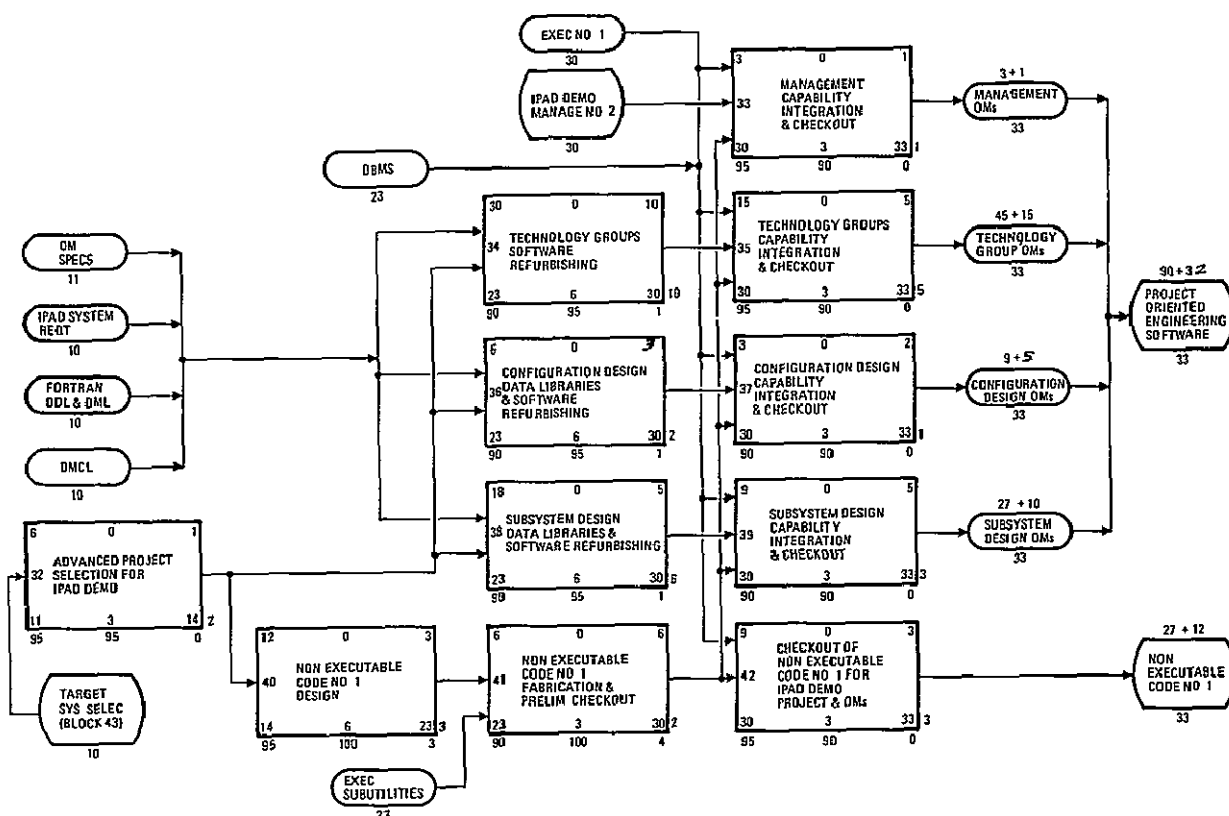


Figure 2-8. Development Phase - Project-Oriented Management/Engineering Software

2.5.2.1 Block 32: The objective of this task is to select an advanced aerospace-vehicle project as a model for an integrated engineering activity during IPAD subsystems and system checkout and demonstration. This task consists of:

1. Approach. An advanced project, preferably in the preliminary design stages, for which a specific set of requirements is available will be selected with NASA concurrence to centralize and amalgamate efforts of the engineering team implementing IPAD, and to make the final selection of OMs to be refurbished. An engineering team composed of

about 11 members will be selected. Disciplines to be represented will be chosen at this time as a function of the type of project selected. The engineering team, when assigned later, will assist in the refurbishing and checkout of the OMs to be used in the IPAD demonstration (Blocks 33 to 42).

2. Manpower, Schedule, and Costs. A team of two senior preliminary design engineers will coordinate with NASA and perform this task in a period of 3 months, with a cost of 6 man-months and one computer hour.

2.5.2.2 Block 33: This task is to integrate the MANAGE No. 2 capability with IPAD EXECutive (reference Section 2.5.4) and with the non-executable code for computing system No. 1. This task consists of:

1. Approach. The intent is to provide an opportunity for detailed checkout of both the OMs and management-related non-executable code prior to subsystem checkout. A secondary intent is to provide preliminary training of project personnel - using the OMs and management-related code - in conjunction with checkout.
2. Manpower, Schedule, and Costs. A project management engineer and a data base analyst will perform this job in 3 months, with a cost of 3 man-months and one computer hour.

2.5.2.3 Block 34: The objective of this task is to refurbish the OMs for the various disciplinary groups represented in the checkout engineering activity. A set of existing engineering capability has been identified in Volume II, Chapter 3 and will be used for this purpose. This task consists of:

1. Approach. The task will be limited to refurbishing those OMs that have a direct use in the checkout and demonstration of IPAD and that are related to the selected project. Many of the OMs will be of general purpose use (e.g., finite element analysis) and, as such, independent of the selected project. The selection of these OMs is simpler and their refurbishing can start early. OMs that are more project dependent must be refurbished with special care to interface with other OMs.

To exploit the interactive features of the IPAD systems while trying to keep engineering software refurbishing costs to a minimum, each OM will be carefully examined to determine the extent it will be refurbished to make it interactive. From this point of view, each selected OM will be classed in one of the following refurbishment categories.

- a. Online plotting - A batch program generates a file of data that is then attached to a general-purpose interactive graphics plotting program. The user may vary the number of data points plotted, rescale the plots, select various coordinate grids, select variables to be plotted, and make comparison plots. However, the batch program must be rerun to change the output data.
 - b. Interactive I/O - Essentially, a batch program is slightly restructured (overlaid to reduce central memory requirements), the ability to change input parameters is added, and the same output is plotted as in Item a.
 - c. Fully Interactive - The program is designed to take full advantage of the new capabilities of the graphic interface. The user can communicate in a graphic or topological way, as well as numerically. New ways of displaying data may be implemented.
2. Manpower, Schedule, and Costs. A team of five disciplinary engineers and five systems and graphics programmers will perform this task in 6 months, with a cost of 30 man-months and 10 computer hours.

2.5.2.4 Block 35: This task is the integration of the refurbished OMs from Block 34 with the EXECutive and non-executable code for computing system No. 1. It is similar to that of Block 33 except for its magnitude (more OMs of a multidisciplinary nature are involved). Schedule and costs are shown in Figure 2-8.

2.5.2.5 Block 36: The objective of this task is to refurbish the automated software available for Configuration Design and to create the data libraries that will be required for checkout and demonstration of the design capability. The task has the following characteristics.

1. Approach. Here again the intention is to refurbish a minimum but adequate set of configuration design software to enable a reasonable engineering team activity for the selected demonstration project. The existing capability in this area and the projected utilization within an IPAD environment was discussed in Volume II, Chapter 3 and will be one of the sources to extract the capability desired for IPAD demonstration.
2. Manpower, Schedule, and Costs. A team of a configuration designer and a system programmer with graphics experience will perform this task in 6 months, with a cost of 6 man-months and 3 computer hours.

2.5.2.6 Block 37: This block is similar to Block 35.

2.5.2.7 Block 38: The objective of this task is to refurbish subsystem design software and create the data libraries that will be required for demonstration. This task consists of:

1. Approach. The subsystem design capability discussed in Volume II, Section 3.2.14 will be used as one source to identify desired software and refurbish it to attain the capability needed for demonstration. Among the data libraries should be a nominal standard parts library, subsystem data bank, specifications and criteria, materials data, and other information needed to demonstrate subsystem design capability. The package will also contain small programs to perform peripheral analysis related to the subsystem design functions.
2. Manpower, Schedule, and Costs. A team of three subsystem designers and three systems and graphics programmers will refurbish this software in a period of 6 months, with a cost of 18 man-months and 5 computer hours.

2.5.2.8 Block 39: This block is similar to Block 37.

2.5.2.9 Block 40: The purpose of this task is to design for computing system No. 1 a Project SCHEMA, task control sequences (TCSs), and TCS skeletons (denoted herein as non-executable code) applicable to the management/engineering/design OMs and the demonstration aerospace-vehicle project. This task consists of:

1. Approach. Extensive data base organization is required before an IPAD system can be installed on a computer. QPSSs must be designed to support many of these requirements, which will subsequently be fabricated using the TCSS/QPSS WRITER, an EXEC subutility (Block 45). A few TCSSs will also be designed. In addition, the proposed designs of a data base will be tested collecting existing DBMS statistics and exploring alternative organizations. The technique of exploring alternative organizations is additionally important to the IPAD Data Base Administrator as well as the specific conclusions determined here. These results will be instructive to IPAD systems installed on all target computing systems.

The target computing system selection (Block 43) will have identified the computing system No. 1 for IPAD's FRC. The design of this non-executable code will specify the code according to the characteristics of the computer system No. 1 (e.g., the host system No. 1 OSCL).

2. **Manpower, Schedule, and Costs.** A data base analyst and a requirements analyst - both extensively knowledgeable of IPAD - will be required for this task. Another data base analyst can also assist by collecting available DBMS statistics on the various target computing systems. The period of performance is estimated at 6 months, with an expenditure of 12 man-months and 3 computer hours. It is not advisable - from cost and coordination standpoints - to subcontract a task so intimately associated with the contractor's design.

2.5.2.10 Block 41: The purpose of this task is to fabricate and conduct a preliminary checkout of the code designed in Block 40. This task consists of:

1. **Approach.** The TCSS/QPSS WRITER developed and checked out in Block 45 will be used to fabricate much of the code in a form adaptable to the requirement of the using project.
2. **Manpower, Schedule, and Costs.** The two data base analysts associated with Block 40 can perform this task in 3 months, with a cost of 6 man-months, and 6 computer hours. Subcontracting of this job is not advisable.

2.5.2.11 Block 42: The objective of this task is to finalize the non-executable code associated with the anticipated IPAD project, the management/engineering/design OMs, and computing system No. 1. Using this code, the OMs and non-executable code will be checked out in an IPAD-like environment. This task will be performed as follows.

1. **Approach.** The intent is to provide an opportunity for detailed checkout of the OMs and project-related non-executable code prior to IPAD sub-systems checkout.
2. **Manpower, Schedule, and Costs.** Two data-base analysts and a systems programmer will perform this task full time for 3 months with a cost of 9 man-months and 3 computer hours.
3. **Risk Assessment.** Programmatic risks are principally associated with the diverse backgrounds of the project personnel and the initial success of the autotutorial approach. The degree to which the OMs or TCSSs (QPSSs) must be altered and/or the project personnel instructed can influence this task. A 90-percent confidence was estimated for schedule and 95 percent for cost.
4. **Alternatives.** Since the principal uncertainty is associated with training project personnel - which is a preliminary to subsystem checkout - the

alternate plan is merely to carry over problems and solve them in this phase.

2.5.3 EXECutive program for computing system No. 1. - The sequence of task Blocks 43 to 47 shown in Figure 2-9 provides the EXEC No. 1 in 30 months from go-ahead, with a cost of 61 man-months and 55 computer hours. This sequence includes the selection of the order in which IPAD will be implemented on the major computing systems, subcontract negotiations with the manufacturer of computing system No. 1, the design, fabrication, and checkout of EXEC No. 1 by the subcontractor, the design, fabrication, and checkout of supporting subutilities, and a final checkout of EXEC No. 1 by the major IPAD contractor. As shown in Figure 2-3, development of the EXECutives for computing systems No. 2 and No. 3 are completed a few months later; i.e., at month 36 for system No. 2 and at month 42 for system No. 3. The following subsections discuss each of the tasks.

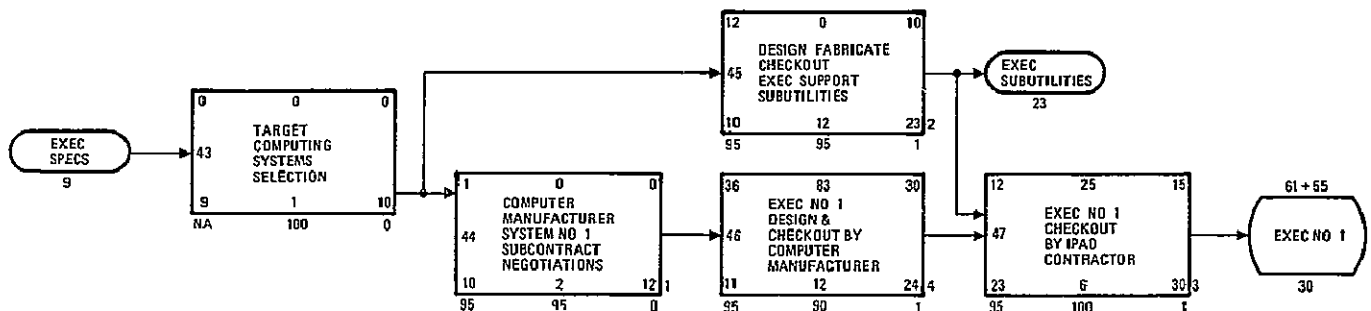


Figure 2-9. Development Phase - EXECutive Program for Computing System No. 1

2.5.3.1 Block 43: The objective of this task is to select the order in which the code associated with the target computing systems, viz:

1. IBM 370/145, 155, 158, 165, 168 with VM/370
2. CDC CYBER 70 (6000 Series) with SCOPE 3.4
3. UNIVAC 1108 or 1110 with EXEC 8

will be prepared for IPAD demonstration on those systems. (See Volume IV, Section 5.5) for a discussion on the target computing systems.) This task will be performed as follows:

1. Approach. Between the eighth and ninth month after contract go-ahead, a final determination must be made as to which computer system will be used first for IPAD demonstration and consequently be available at FRC. This selection is necessary to ensure orderly development of the required

code to support these systems. Staggered development of this code is reflected to reduce peak funding and to minimize the contractor's IPAD manpower-loading fluctuations. Several factors contribute to the final selection:

- a. The degree to which the target system is capable of supporting IPAD, viz:
 - The degree of their support to the COBOL DBMS capability and the recently developed DMCL (Block 23).
 - The stability and capability of their graphics code (required to support the general graphics library).
 - The stability of their current operating systems (e.g., has it recently undergone or is it currently scheduled to undergo a major change?).
- b. A NASA request for PRC or FRC to support a specific system in a specific time frame.
- c. The computers immediately available to the contractor, since this can affect schedule/costs and performance on the contract.
- d. The number of expected users of IPAD on the various systems.

This decision should be delayed as long as possible to facilitate the correct choice.

2. Manpower, Schedule, and Costs. This is an administrative decision to be made by NASA/LRC based on data collected by and recommendations from the contractor. No costs are associated with this task. One month is more than adequate.
3. Risk Assessment. The principal technical risks are a decision based on obsolete data or a decision rendered obsolete by subsequent developments.
4. Alternatives. The alternatives in either case are:
 - a. Continuously monitor the decision factors affecting system selection.
 - b. If it appears likely that a change in selection is required, prepare an alternate plan at that time and assess the impact and elect the least

schedule, cost, or risk approach (as directed by NASA LRC).

It is the nature of competition that business decisions can be altered literally overnight, often negating a prior commitment. There is usually no choice but to adapt to this change.

2.5.3.2 Block 44: The objective of this task is to negotiate with the manufacturer of computer system No. 1 to fabricate the IPAD EXEC, exploiting the available, standard features of his operating system and reach agreement and award a subcontract. This task consists of:

1. Approach. The various computer manufacturers are the best candidates to construct the portion of the IPAD EXEC that must interface closely with their systems because:
 - a. They are the most knowledgeable concerning their systems, often having access to specific persons who fabricated the interfacing code.
 - b. They are generally the most knowledgeable of pitfalls to be avoided, compromises to be made, and decision factors to be evaluated.
 - c. They alone are knowledgeable of the direction that their system might take and can factor this into their design.
 - d. If they are responsible for the design and fabrication of the interface code and if they adhere to exploiting only standard, available features, there is:
 - Less likelihood that a subsequent system change will render the EXEC inoperative.
 - A higher probability that the manufacturer will eventually accept responsibility for the code.
2. Manpower, Schedule, and Costs. A senior computer systems engineer will be in charge of the technical aspects of this negotiation. A time span of 2 months with a cost of one man-month are estimated for this task.
3. Risk Assessment. The technical risks associated with this task are:
 - a. The manufacturer is not interested in the task.
 - b. The manufacturer, although interested, will not negotiate to do the task for what appears to be a reasonable price.

- c. The manufacturer disagrees with the technical aspects of the task.
4. Alternatives. The first two risks necessitate direct negotiation with other qualified firms (e.g., software houses) to attempt to place the task. This could result in placing the task or the awareness that the "reasonable price" is indeed too low. If the price is too low, there is little recourse but to accept the increased cost and proceed with the task. Unless there are major differences in the price quotes, the manufacturer would be selected as the subcontractor for the reasons discussed in Item 1. above.

The third technical risk is the most serious, since it affects the design requirements of the EXEC on that manufacturer's computer and possibly on the other computers as well (i.e., affecting the basic design of IPAD's EXEC). The circumstance can be avoided by coordinating the design of the EXEC (Blocks 7 and 8) with the manufacturers well in advance of sub-contract negotiations.

2.5.3.3 Block 45: The objective of this block is to design, fabricate, and check out subutilities supporting IPAD's EXEC, which are applicable to all computing systems. The characteristics of this task are:

1. Approach. There are four FORTRAN subutilities supporting IPAD's EXEC:
 - a. TCSS/QPSS WRITER. This is an interactive program that assists an experienced user in writing a task control sequence skeleton (TCSS) or a query processor session skeleton (QPSS). Input to the program is usually a TCS or QPS that addresses a specific task; these could have been in successful previous use within the system or been specially constructed with the host system's text editor. Optional input is a textual description of the intended use (objectives) of the TCSS (QPSS). The interactive writer will enable the user to tag items within the TCSS (QPSS) with assigned external reference names, compile a set of pointers, compile a glossary of these names, compile a set of tutorials, generate (or use the optionally input) textual description of the TCSS's (QPSS's) intended use, and finally construct the TCSS (QPSS) and catalog it within the system.
 - b. TCSS/QPSS EXPANDER. This is an interactive (or batch) program that assists an inexperienced user in tailoring a TCS (QPS) - from a general TCSS (QPSS) - for his specific usage. The textual description should inform him of its intended use so he can determine that he has the proper TCSS (QPSS). The tutorials and name glossary will guide him through the assignment of particulars to the names. Once complete,

the EXPANDER will produce the required TCS (QPS) file and, optionally, execute it. (It may be an executed batch with serial name assignment.)

- c. INTERCEPTOR. This is a subutility directly supporting the EXEC in recording the user's TCSs (QPSs) as they are executed which:
 - Enables the user to maintain a record of his transactions — TCSs and QPSs — as actually used in sequence.
 - Provides input for the user task trajectory (UTT) if one is requested by the user's supervision.
- d. UTT RECORDER. This is a batch subutility that post-processes the input file created by the INTERCEPTOR (if one was requested) and creates an extension to the user's UTT file.

These subutilities are described in Volume V, Part I, Sections 2.2 and 2.3. The EXPANDER and RECORDER are the principal tasks; the other two are trivial in comparison.

The approach is to finalize the design and to fabricate and conduct a preliminary checkout of this code in parallel with the development of IPAD's EXEC No. 1 (Block 46). The task includes fabrication of some general use TCSSs.

2. Manpower, Schedule, and Costs. Two senior programmers with experience in generating efficient FORTRAN code will perform this task in 12 months, with a cost of 12 man-months and 10 computer hours. No subcontracting of this task is contemplated.

2.5.3.4 Block 46: The objective of this task is to design, fabricate, and check out (in a preliminary fashion) the host operating system interface code and the core of IPAD's EXEC on computer system No. 1, as subcontracted in Block 44. This task will be accomplished as follows.

1. Approach. A contractor-supplied system programmer/analyst will work full time with the subcontractor during the design and checkout phases of the EXEC and will monitor the fabrication phase at periodic intervals. Since the programming language used will probably be the machine's assembly language, the contractor's system analyst — who is to become one of the product demonstration team members — must understand the functions ascribed to each code module. The contractor's system analyst

will retain a copy of the code produced and will be qualified to modify this code as required.

2. **Manpower, Schedule, and Costs.** In addition to the contractor's system programmer analyst, the subcontractor will supply a system programmer/analyst to work directly with the contractor's analyst during the design and preliminary checkout phases. This subcontractor's analyst will also be contracted for the extensive EXEC checkout phase (Block 47), the IPAD subsystem checkout phase, and the IPAD demonstration phase. The subcontractor will also supply system programmers and specialists as required to complete the task.

For schedule and cost estimates only, it was presumed that the computing system requiring the most extensive interface design could be selected as system No. 1. The duration of this task is estimated at 12 months, with a cost of 36 man-months and 30 computer hours. These costs could be reduced by likely system improvements provided by the computer manufacturers, although no specific plans are known at this time. The 83 percent subcontract factor provides six man-months for the contractor-supplied system analyst. This will principally be used during the design and checkout phases.

3. **Risk Assessment.** Due to earlier coordination with the target computing system manufacturers and the recently completed subcontract negotiations with the manufacturer of system No. 1, there will be a clear understanding of the technical aspects of the job. The only risk that can be identified is that the subcontractor fails to perform to schedule and/or cost.
4. **Alternatives.** The impact of schedule delays could probably be absorbed by the slack time available. A cost overrun, if any, must be evaluated and negotiated according to the circumstances.

2.5.3.5 **Block 47:** The objective of this task is to provide an extensive checkout of IPAD's EXEC No. 1 in an IPAD-like operating environment. The features of this task are:

1. **Approach.** IPAD's EXEC No. 1 and its support subutilities will be installed in a scientific operating environment on host computing system No. 1. This code will be exercised extensively at various times throughout the computing day and with various other subsystems to assess the response and reliability under different operating system work loads. Since the EXEC's EXPANDER, WRITER, and INTERCEPTOR subutilities are very useful in a conventional, non-IPAD environment, attempts will

be made to expand the use of these subutilities for conventional batch and interactive task processing as a further test of their efficacy and stability.

2. **Manpower, Schedule, and Costs.** The contractor's system programmer/analyst and the principal subcontractor's system programmer/analyst from Block 46 will join the programmer from Block 45 to form the EXEC No. 1 checkout team. It is anticipated that the programmer from Block 45 will conclude his task quickly and retire from the team. Although the team will start out full time, the task is envisioned to conclude with the EXEC being exercised on an operational basis using the EXEC's INTERCEPTOR and RECORDER subutilities to prepare summaries on the types and frequency of transaction being conducted via the EXEC. Six months are estimated for this task, with a cost of 12 man-months and 15 computer hours. The 25 percent contract factor is for the services of the subcontractor's system programmer/analyst.
3. **Risk Assessment.** The programmatic risks are mainly associated with unexpected coding errors or deficiencies that could become apparent during this extreme checkout and that could influence the estimated costs.
4. **Alternatives.** No alternatives to this task were identified. Code errors and deficiencies must be corrected to provide a functional EXECutive code.

2.5.4 Data base management system (DBMS).— Figure 2-10 shows the task sequence that will provide a DBMS capability 23 months after go-ahead, with a cost of 76 man-months and 31 computer hours. Previous tasks that have a bearing on the development of the DBMS capability are the IPAD system requirements, the Fortran Temporary Code specifications, the working documents produced in Block 19, and the FORTRAN DDL and DML documents.

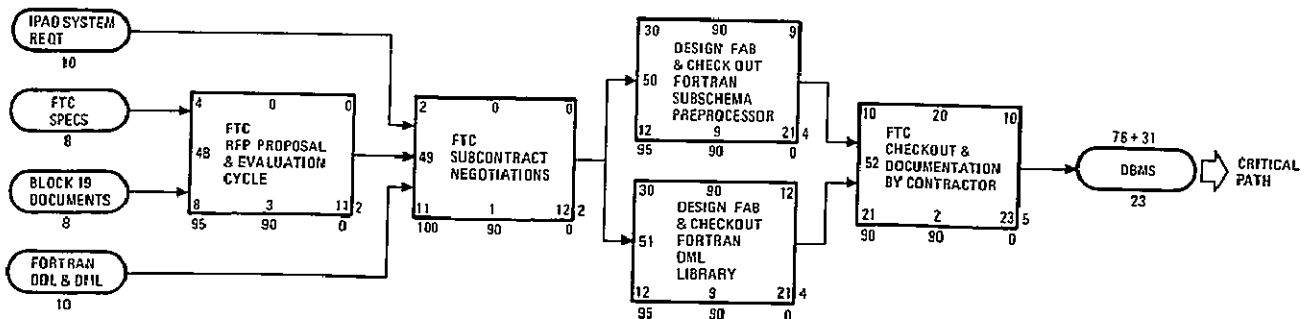


Figure 2-10. Development Phase - Data Base Management System

This development plan presumes the worst case from an IPAD implementer's point of view, viz., that the FORTRAN DML takes the form of a language extension to FORTRAN (e.g., new verbs logically replacing READ and WRITE). What is required, then, is:

1. An extension to the DML Insertion Preprocessor (a Special Purpose Utility, see Section 2.5.7) to optionally replace conventional FORTRAN I/O with equivalent:
 - a. FORTRAN DML
 - b. FORTRAN DML CALL statements replacing the DML verbs.
2. A FORTRAN DML library to field the DML CALLs and satisfy these via equivalent COBOL DML. (This library - except possibly the interface with the CALLing FORTRAN subprograms - would logically be written in COBOL).
3. A preprocessor to map the logically equivalent FORTRAN SUBSCHEMA (SCHEMA) from FORTRAN DDL to COBOL DDL. The existing DDL compiler could then compile the SUBSCHEMA (SCHEMA).

This plan is contingent upon the recommendations of the FORTRAN DML/DDL task group. If this task group recommends using FORTRAN CALLs to interface with DBMS and COBOL DDL to write the FORTRAN SUBSCHEMA (see Subsection 2.4.6.2 for an applicable discussion), the subplan reduces to simply supplying the FORTRAN DML library. Details on the various tasks are given in the following subsections.

2.5.4.1 Block 48: The purpose of this task is to write a request for proposal (RFP) and evaluate the responses from computer manufacturers and software developers relative to the design, fabrication, and checkout of the FORTRAN Temporary Code (FTC). This task will be performed as follows:

1. Approach. There are two utilities associated with FTC:
 - a. FORTRAN SUBSCHEMA preprocessor.
 - b. FORTRAN DML library.

Consequently, two separate RFPs will be prepared and separate bids will be sought. Technical evaluation will be based principally on a deep understanding and appreciation for the problem and on the bidder's past experience in the field. An additional factor in the evaluation will be their

management plan to ensure meeting schedule/costs with consistent high quality. The best designer/fabricator for the most realistic cost will be selected for the development (with the concurrence of NASA).

It is anticipated that only computer manufacturers and software developers will bid on the FORTRAN SUBSCHEMA preprocessor due to the requirement to demonstrate past (applicable) experience. However, firms with a moderately large business programming group (and access to FORTRAN system analysts) could also bid on the FORTRAN DML library. There should be no lack of qualified bids.

Although the contractor could absorb the FORTRAN DML library if required, both tasks are to be placed with qualified subcontractors if possible.

The RFPs will be written using the in-process documents of the FORTRAN DDL/DML Task Group from Block 19. The final documents from Block 20 will be used in subcontract negotiations discussed in Block 49.

2. Manpower, Schedule, and Costs. A team comprising a data-base analyst and a computer system programmer/analyst will write the technical aspects of the proposals and provide technical evaluations of the responses. Three months are estimated for the duration of this cycle with a cost of 4 man-months.

2.5.4.2 Block 49: The objective of this task is to award the FTC subcontract(s) and to conduct the pertinent negotiations. It is assumed that two subcontractors will be involved, with the possibility of a single subcontractor for both developments.

Since the FORTRAN DDL/DML task group's recommendations were preliminary at RFP issue time and are very likely to change (moderately rather than radically), it will be necessary to include these changes, if any, in the final contract negotiations. This is indicated by the final FORTRAN DDL/DML documents feeding directly into Block 49.

This task will require one month calendar time and an estimated cost of 2 man-months.

2.5.4.3 Block 50: The objective of this subcontracted task is to design, fabricate, and check out the FORTRAN SUBSCHEMA Preprocessor. Specifically, this code will map FORTRAN DDL onto equivalent COBOL DDL. This task will be conducted as follows.

1. Approach. The subcontractor, having first obtained approval of the detailed design of the preprocessor, will fabricate and conduct a preliminary checkout of the code using hand techniques (viz., examining the preprocessor replacement to see that the correct substitutions were made). The contractor will review the design and monitor the task at periodic intervals.
2. Manpower, Schedule, and Costs. The subcontractor must provide a compiler programmer and various systems programmers. A main contractor data-base analyst will perform the technical reviews of the subcontracted work. Due mainly to schedule constraints, this task is estimated to cost 30 man-months plus 9 computer hours and to be performed during a period of 9 months.
3. Risk Assessment. The identified technical risks are:
 - a. Unforeseen technical difficulties.
 - b. Inadequate performance by the subcontractor.

Both risks are complicated by the fact that this task is on the critical path.

4. Alternatives. The only alternative to unforeseen technical difficulties is to accept the schedule/cost impact. Two realistic options are available for inadequate performance:
 - a. Cancel the subcontract and absorb the task.
 - b. Alter the subcontract and take over technical direction (in addition to the technical management) of the task.

2.5.4.4 Block 51: The objective of this task is to design, fabricate, and check out (principally) COBOL code that will provide DBMS support to FORTRAN DML. This task will be performed as follows.

1. Approach. The subcontractor, having first obtained approval of the detailed design of the library, will fabricate and conduct a preliminary checkout of the code, using programs supplied by the contractor and a host computing system (perhaps the contractor's) providing DBMS support to COBOL. The contractor would additionally review the design and monitor the task at periodic intervals.

2. **Manpower, Schedule, and Costs.** The subcontractor will provide a business system programmer experienced at COBOL data-base work as well as system programmers. The contractor shall assign a system programmer/analyst for technical review. This task is scheduled for 9 months at a cost of 30 man-months and 12 computer hours.
3. **Risk Assessment.** Risk is considered identical to that of Block 50.
4. **Alternatives.** The alternatives are the same as those for Block 50, except that the main IPAD contractor is more likely to absorb this task.

2.5.4.5 **Block 52:** The objective of this task is to provide detailed checkout of the code supplied in Blocks 50 and 51 to ensure trouble-free DBMS support to FORTRAN and to make corrections as required and provide documentation of the code. This task consists of:

1. **Approach.** The contractor will undertake a two-month detailed checkout of FTC to detect and correct the errors or omissions in the supplied code. For this purpose, special FORTRAN test cases will be fabricated from existing code modules. Documentation supplied by the subcontractors in Blocks 50 and 51 will be reviewed for completeness.
2. **Manpower, Schedule, and Costs.** The main contractor personnel that performed the technical reviews of the subcontracts of Blocks 50 and 51 will conduct the checkout assisted by another system programmer and representatives of the engineering user community. A period of two months is assigned to this task, with a cost of 10 man-months and 10 computer hours.
3. **Risk Assessment.** A flaw in the code produced in Blocks 50 and 51 should quickly become apparent in Block 52. It is anticipated that only minor errors will be discovered and the remaining time spent in further checkout (to establish confidence) and in refurbishing the documentation. If substantially more flaws are detected, additional manpower could be called upon to rectify the code, possibly drawing from the original developers (the subcontractors). The cost confidence (90 percent) reflects this risk. Although this could adversely affect the degree of checkout accomplished, it is very unlikely to alter the schedule.
4. **Alternatives.** There are no alternatives to this task.

2.5.5 General graphics library (GGL). - Figure 2-11 shows the task sequence required to develop a GGL for each of the three target computing systems. This sequence is completed 23 months after go-ahead, with a cost of 73 man-months and 40 computer hours. There are five distinct libraries of code associated with the GGL:

1. A FORTRAN library handling the direct view storage tube (DVST) terminals (e.g., the Tektronics 4010).
2. A FORTRAN library handling the hardcopy devices (e.g., CALCOMP).
3. A separate library for each of the three target computing systems (identified as GGL No. 1, 2, and 3). These libraries consist of the FORTRAN-FORTRAN interface required to provide the minimum GGL support (Block 12) on the host computer. The code fields requests in GGL and satisfies these by making FORTRAN CALLs to the host system's graphics library (e.g., UNIGRASP in the case of UNIVAC's EXEC 8).

The Baseline Implementation Plan assumes that the DVST and hardcopy device libraries will be developed by the IPAD contractor, while GGL No. 1, 2, 3 will be developed under subcontract. Details on each of these task blocks are given in the following subsections.

2.5.5.1 Block 53: The purpose of this task is to write a request for proposal (RFP), evaluate the responses, and award subcontracts relative to the development of GGL No. 1, 2, and 3. The task will be performed as follows.

1. Approach. A single RFP will be written using the GGL specifications, but separate responses will be sought for each target computing system. There are numerous firms that could bid on these tasks. Any of the target computer manufacturers could handle their graphics library interface with ease. There should be no lack of qualified bids, many of which should fall within a relatively narrow price range.

Due to the straightforward nature of this task, the contractor could absorb any of these tasks if required. Indeed, the contractor may elect not to issue the RFP for interface to his own system in the event of a schedule stretchout, or if required to balance manpower loading.

2. Manpower, Schedule, and Costs. A graphics programmer/analyst assisted by an engineering user will write the technical aspects of the proposals and provide technical evaluations of the responses. The duration of this cycle is estimated at 3 months, with a cost of 4 man-months.

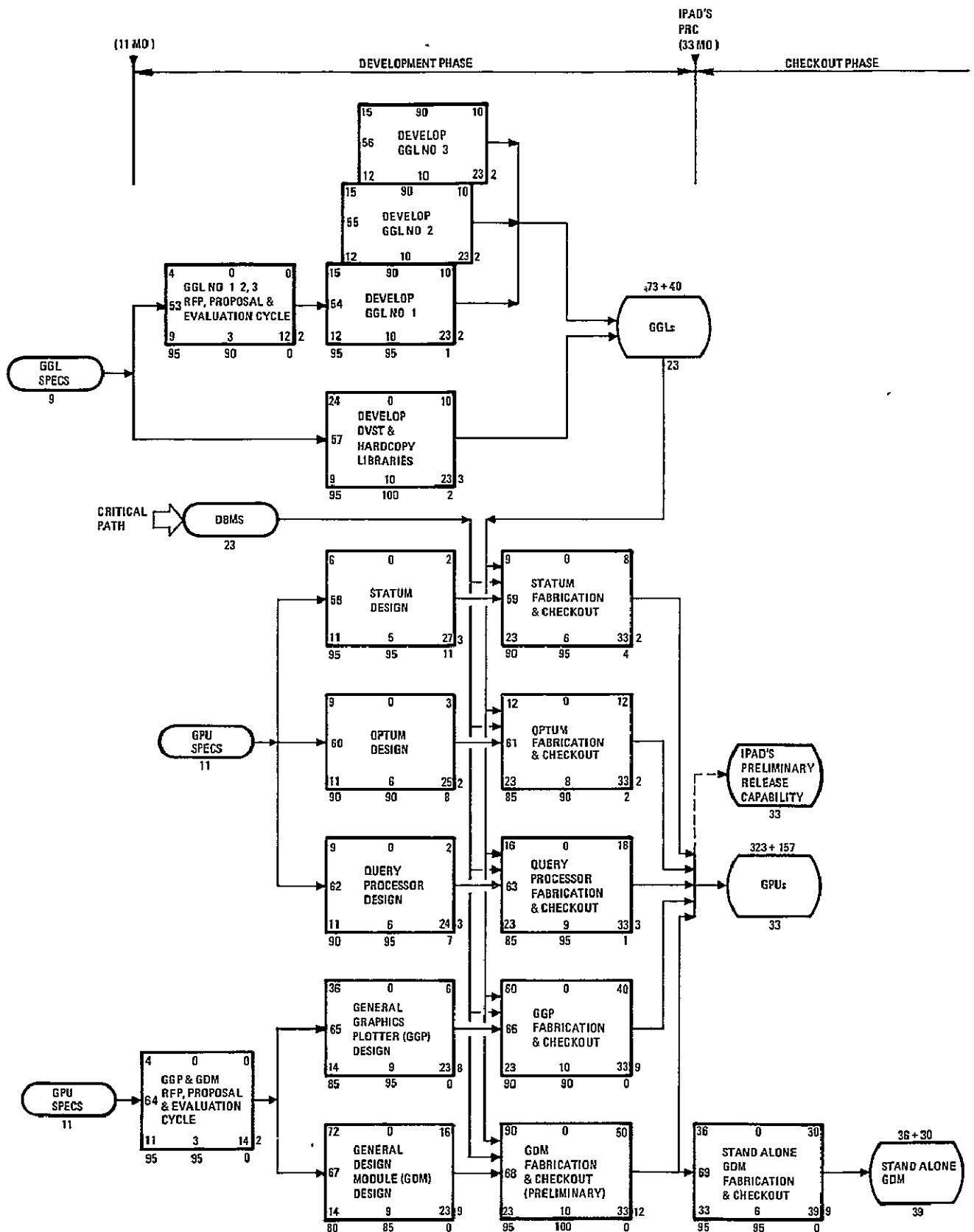


Figure 2-11. Development Phase - GGLs, GPUs, and Stand-Alone GDM

2.5.5.2 Block 54: The objective of this task is to develop a GGL for interfacing the graphics library of one of the target host computers (here termed GGL No. 1). The task consists of.

1. Approach. A considerable amount of code is available to support the GGLs. For the host system support, each computer manufacturer has a library supporting his refreshed CRT terminals. The developmental approach is to divide the task into a design phase and a fabrication/checkout phase. Fabrication can begin immediately following approval of the detailed design, with checkout occurring incrementally as modules are completed. The task concludes with final product demonstration at the subcontractor's facility using the checkout code developed for this purpose by the contractor.
2. Manpower, Schedule, and Costs. The subcontractor shall provide graphics and systems programmers to handle the mapping of GGL onto the host language. A contractor-supplied graphics analyst is required to review the GGL No. 1 library to ensure compliance with the design requirements. This task is scheduled for 10 months, with a cost of 15 man-months and 10 computer hours. A 90-percent subcontract factor provides six man-weeks for review of the design and checkout of this library.
3. Risk Assessment. The programmatic risks are assumed to be minimal, since the task is straightforward and more than adequate time is available to plan and execute this task.
4. Alternatives. In case of inadequate performance by the subcontractor, the alternative is to cancel the subcontract and absorb the task.

2.5.5.3 Blocks 55 and 56: These two tasks are identical to Block 54, except that they are for GGL No. 2 and GGL No. 3 respectively.

2.5.5.4 Block 57: The objectives of this task are to develop a library for the direct view storage tube (DVST) class of graphics terminals and a hardcopy library to support such widely used devices as the SC-4020, CALCOMP, and GERBER. This task consists of.

1. Approach. The manufacturers of DVST terminals and of hardcopy devices support their products with appropriate code. Many of these devices are CALL-compatible with the SC-4020 microfilm recorder library of subroutines originally developed by North American Aviation in the late 1950s. Indeed, there are many CALL-compatible FORTRAN libraries that service a wide variety of hardcopy-device types. An example of such a library (built around a CALCOMP plotter) is NASA-LRC's Graphic Output System which services (quoting from Reference 2):

"... the CALCOMP and EAI 430 Electro-Mechanical Plotters, [and also] the Gerber Drafting System and the DDI 80B and CDC 252 Electronic-CRT Plotters.

"The advantage of device-independent graphic software is that a programmer then has the option of choosing the display device which can best satisfy the job requirements. A further advantage is that the chosen output device may be varied easily with no significant change in the format of the display."

As an example of manufacturer support, Tektronics' PLOT-10 is GGL-like DVST capability supporting Tektronics' 4000 series DVSTs. (Volume V, Part I, Section 5.6 contains a detailed discussion of these libraries.)

2. Manpower, Schedule, and Costs. A team of three systems programmers will complete this task in 10 months, with an expenditure of 24 man-months and 10 computer hours. These costs are justified on the basis that:
 - a. There are at least a half-dozen devices in each category to be considered.
 - b. Each device has somewhat different capabilities.
 - c. The DVSTs, having less capability than the host computer's refreshed CRTs, must circumvent some of inapplicable GGL (e.g., "blinking") and substitute for others (e.g., "selecting" replacing "picking").

2.5.6 General purpose utilities (GPU) and stand-alone general design module (GDM). - This sequence of tasks provides these elements of software 39 months after go-ahead, with an expenditure of 323 man-months plus 157 computer hours for the GPUs and 36 man-months plus 30 computer hours for the stand-alone GDM. The GPUs include STATUM, OPTUM, the Query Processor, the General Graphics Plotter (GGP), and the GDM. In addition, a stand-alone version of GDM supported by a minicomputer is provided after completion of the GPUs. Included in this plan are provisions to subcontract a portion of the efforts to obtain GGP and GDM. Details on each of these tasks are given in the following subsections.

2.5.6.1 Block 58: The objective of this task is to design — preparatory to fabrication — the STATistical Utility Module (STATUM) and to review STATUM prior to fabrication/checkout. This task will be accomplished as follows.

1. Approach. The intended design of STATUM is thoroughly treated in Volume V, Part II, Section 2. Since most of the required code already exists, the design principally addresses the problem of locating applicable existing code and supplying the required interactive interfaces.

2. Manpower, Schedule, and Costs. A computer systems programmer with assist from another programmer and an engineering user will provide the design of this utility in a period of 5 months, with an expenditure of 6 man-months and 2 computer hours.

2.5.6.2 Block 59: This task is to provide fabrication and checkout of the STATUM utility designed in Block 58 and to interface it properly with the DBMS and the GGL software. The task consists of:

1. Approach. The approach is to check out code modularly in conjunction with a fabrication and to conduct periodic technical reviews. By final product demonstration, the code will have met all design requirements.
2. Manpower, Schedule, and Costs. A programmer and an engineering user can complete this task in 6 months, with a cost of 9 man-months and 8 computer hours.
3. Risk Assessment. The major technical risks are slippage of the DBMS support software or of the DBMS support to COBOL on the contractor's computer, or of the GGL support software. The latter circumstance is highly unlikely, since GGL is a low-risk development and would have to slip over four months to have any appreciable impact.

2.5.6.3 Block 60: The objective of this task is to design — preparatory to fabrication — the general-purpose OPTimizer Utility Module (OPTUM). This task is similar to that of Block 58, except that a data-base analyst is required to interface the design with DBMS. The schedule and costs are as shown for Block 60.

2.5.6.4 Block 61: This task is to fabricate and check out the code generated in Block 60 and properly interface it with the DBMS and the GGL software. This code is more sophisticated than that for STATUM and will require more effort. Further, the data-base analyst of Block 60 will be required for consultation and review. A duration of 8 months is contemplated, with an expenditure of 12 man-months and 12 computer hours.

2.5.6.5 Block 62: The objective of this task is to design — preparatory to fabrication — the Query Processor (QP), which provides for user control of his data sub-bases. To review QP prior to fabrication/checkout.

QP will represent perhaps the most widely used IPAD GPU. It is critical to the function of IPAD and merits being carefully designed and checked out. This task consists of:

1. Approach. QP differs from the other GPUs in that it is best written in COBOL and its sole purpose is to manipulate the contents and structure of various data sub-bases. The required functional interface is the QPL developed in Block 15. The task is somewhat complicated by the preliminary nature of the DMCL. (See Blocks 21, 22, and 23.)

QP was originally envisioned to be a manufacturer-supplied utility supporting the data base via their implementation of DBMS. Indeed, most manufacturers will have a QP-like utility in their DBMS product subset. However, the costs and technical risks associated with

- a. Developing a QPL that could become a "de facto" standard, and
- b. Getting the major computer manufacturers to adopt QPL and provide the utility when required by IPAD

were considerable more than to provide a separate IPAD utility. Further, QP presents little if any difficulty to bring about, particularly in COBOL. Consequently, QP was added to the GPUs and became an IPAD utility which duplicated — to a certain extent — code planned by the manufacturers.

It is envisioned that this task will be accomplished by a business programmer experienced in COBOL DML/DDL. The design task is to provide a map of GGL onto COBOL DML and an efficient technique for execution.

2. Manpower, Schedule, and Costs. One business systems programmer, skilled in COBOL data-base management, can handle this task with a data-base analyst and requirements analyst for consultation/review. A performance period of 6 months is estimated, with a cost of 9 man-months and 2 computer hours.

2.5.6.6 Block 63: This task is to fabricate and check out the code designed in Block 62 and to include interfaces with the DBMS and GGL software. Although the code sophistication is not greater than that of OPTUM, a larger effort is required to check it under more numerous and varied engineering user circumstances. The team that performed task Block 62 will participate in this task, assisted by an engineering user. A duration of 9 months is estimated, with a cost of 16 man-months and 18 computer hours.

2.5.6.7 Block 64: The purpose of this task is to procure subcontracted efforts on portions of the General Graphics Plotter (GGP) and the General Design Module (GDM). It is difficult at this time to identify what portions will be subcontracted, since this is a dynamic field of activity by various aerospace companies, software developers, and

government agencies. These independent efforts will have a definite impact on the extent of new modules that will be required to complement the software available at that time, thus providing the sought-after capability. The GGP and GDM fill the fundamental needs of the project configuration and subsystems designers. It is important that field-proven modules already in use at various aerospace companies be incorporated into the GGP and GDM capabilities. Royalties and buy prices may be associated with some existing elements or packages of software presently being marketed within industry.

The purpose of the proposed buy/subcontract approach is to allow participation of the best know-how available in industry at large and to incorporate into IPAD the most advanced elements and/or packages of software available, on a cost-effective basis. This procurement task consists of:

1. Approach. Two independent RFPs will be prepared (one for GGP, the other for GDM) using the GPU specifications previously developed and the latest results of a continuous survey of developments in these fields.

Technical evaluation of the submitted proposals will be based on thorough understanding of the tasks, previous bidder's applicable experience, and soundness of advanced ideas. The management plan to ensure meeting schedule and cost with high product quality will be an important consideration.

2. Manpower, Schedule, and Costs. The technical aspects of this proposal will be written by senior system and graphics programmers assisted by engineering users and IPAD study management personnel. The same team will provide technical evaluation of the proposals submitted. A duration of 3 months is estimated for this task, with a cost of 4 man-months.

2.5.6.8 Block 65: The objectives of this task are to design — preparatory to fabrication — the General Graphics Plotter (GGP) and to review GGP prior to fabrication/checkout. This task consists of:

1. Approach. The intended design of GGP is thoroughly treated in Volume V, Part II, Section 5 and further illustrated in Section 6. Its purpose is to provide:
 - a. A general plotting capability for data items within the data base (viz., both graphic and pictorial plotting).
 - b. A general capability to organize and display topological entities and record this organization within the data base.

GGP draws heavily on all capabilities provided by GGL and DBMS. It is probably the most sophisticated of the IPAD utilities, the next largest (behind GDM), and the third most frequently used (behind GDM and QP).

Finalization of the subcontracted portions of GGP should be accomplished by the subcontractor, with frequent reviews by the contractor's technical team.

2. **Manpower, Schedule, and Costs.** A team of eight people including senior graphics and systems programmers and engineering users from both the contractor and subcontractor(s) will be required for this task. A period of 12 months is available for the main IPAD contractor, while the subcontractor(s) efforts must wait for completion of the procurement cycle and are estimated to start 14 months after go-ahead. The expenditures for this task are 36 man-months and 6 computer hours. The buy items of software required for GGP are assumed to be included in the estimated total costs of Blocks 65 and 66.
3. **Risk Assessment.** The technical sophistication of this task will call for close management and coordination to determine:
 - a. The exact status of a given subtask.
 - b. The existence of a problem and its nature (technical or administrative).
 - c. Accurate forecasts to complete.

This increases the risks of schedule slippage and cost overruns. The confidence factors shown for cost and schedule reflect this basic problem and presume diligent monitoring of the task with respect to the plan.

4. **Alternatives.** The modularity of GGP will permit the contractor to absorb the development of certain modules in case of poor performance by the subcontractor(s).

2.5.6.9 Block 66: The purpose of this task is to fabricate and check out GGP for which design has been approved in Block 65. This task consists of:

1. **Approach.** Once the design has been completed and approved, fabrication and checkout of the code is somewhat complicated by the requirement to interface with the DBMS and GGL support software. Since GGP draws heavily on all capabilities provided by GGL and DBMS, any slippage of or excessive difficulties with the support software could slip GGP, which is on the critical path.

The approach is for the contractor and the subcontractors to check out code modules in conjunction with fabrication and to provide periodic reviews of progress. Detection and early resolution of problems that may occur with the support software is critical to this task.

2. Manpower, Schedule, and Costs. The same contractor and subcontractor teams will be assigned to this job. Due to the sophistication of this task and the critical schedule of 10 months, a cost of 60 man-months and 40 computer hours was estimated.
3. Risk Assessment. The technical risks identified with this task are:
 - a. Slippage of the DBMS support software.
 - b. Slippage of the DBMS support to COBOL on the contractor's or subcontractor's computer.
 - c. Slippage of the GGL support software.
 - d. Inadequate subcontractor performance.
4. Alternatives. The only alternative to the first risk is to accept schedule slippage, with some impact on costs. The second risk can be circumvented by providing module checkout and product demonstration at another facility. This will increase schedule/costs due to the necessity for:
 - a. Travel to or teleprocessing into the other facility.
 - b. Training of the contractor and/or subcontractor personnel on the computer at the other facility.
 - c. Scheduling of computer time at the other facility.

The third risk is less likely to occur due to the high schedule confidence attached to the development of GGL.

The unlikely fourth risk — that of inadequate subcontractor performance — is complicated by reluctance of the contractor to absorb this task at this point in time, but it may be the only alternative left.

2.5.6.10 Block 67: The objective of this block is to design — preparatory to fabrication — the core elements of the General Design Module (GDM) and to review this design prior to fabrication/checkout. This task consists of:

1. **Approach.** The intended design of GDM is outlined in Volume V, Part II, Section 7. Its purpose is to provide the tools to be used by the board designers in an IPAD environment. GDM draws heavily on all concepts of GGL and DBMS, but builds these as support software on the minicomputer. It is destined to be the largest and most frequently used of the IPAD utilities. Due to the requirement for very fast response time, GDM is to be designed to run only on a minicomputer; subtask delegation by the minicomputer to the host computer will be a relatively infrequent occurrence. This requires that a data-base management subsystem — similar to but less sophisticated than DBMS — be constructed for the minicomputer. Further, a GGL must be fabricated to support graphics code on the minicomputer.

GDM involves the development of large amounts of new code. The approach taken is unlike the other GPUs in that fabrication of the core elements of this code will not be held up pending a completed detailed design of GDM.

2. **Manpower, Schedule, and Costs.** A team of nine members from the contractors and subcontractors shall include design/engineering users, mini-computer system programmer/analysts, graphics programmers, data-base analysts, and applications programmers. A period of 12 months is available for the main contractor's efforts, which do not have to wait for the RFP cycle to be completed; the subcontractor's development studies will commence 14 months after go-ahead (when subcontract negotiations have been completed). The cost of this task, inclusive of subcontracts and equivalent buy item prices, is estimated at 72 man-months and 16 computer hours.
3. **Risk Assessment and Alternatives.** The discussion given in subsection 2.5.6.8 is totally applicable to the GDM.

2.5.6.11 **Block 68:** The objective of this task is to fabricate and check out the core modules of GDM and selected portions of additional GDM capability deemed appropriate for GPU subsystem checkout and an IPAD Pre-Release Capability (PRC). This task will be performed as follows.

1. **Approach.** Unlike the other GPUs, GDM will overlap fabrication/checkout with design. This is necessary because the total task duration is too long to accommodate IPAD schedules without a six-month stretchout. This is true only for GDM.

The plan is to provide an interim GDM for subsystem checkout (and for PRC) and continue to complete and subsequently check out a stand-alone capability

on the minicomputer (task Block 69). Since the majority of GDMs code resides on a minicomputer, problems within the DBMS and GGL support software will have a minor effect on this task.

2. **Manpower, Schedule, and Costs.** The same contractor and subcontractor team identified for Block 67 will continue with the task, augmented by one or more programmers and engineering users, as required. The same contractor's review team will continue periodic monitoring of the design and code fabrication efforts. Code modules for subsystem checkout must be delivered within 10 months. These must include the principal support modules on the maxicomputer as well as enough minicomputer code to exercise the mini-maxi-mini loop. The schedule merely provides a delivery constraint on this software. The cost of this task is estimated at 90 man-months and 50 computer hours.
3. **Risk Assessment.** Since the precise amount of code delivered is not critical to subsystem checkout, code modules beyond the minimum requirements can be selected well in advance. Correspondingly, there is no schedule risk (100-percent confidence). Further, since the intent is to hold the level of activity reasonably constant during this phase, the programmatic risk associated with cost is minimal (95-percent). Technical risks are identical to those identified in Block 66 for the GGP, except that the slippages related to DBMS and GGL are much less likely to affect this task.

2.5.6.12 **Block 69:** The purpose of this task is to continue GDM fabrication and checkout, culminating in a stand-alone capability on the minicomputer and a completion of the mini/maxi interface. The task consists of:

1. **Approach.** The task calls for a finalization of the GDM design that will support IPAD's First Release Capability (FRC) and the fabrication and checkout of the supporting code. Where required, the final design of GDM will be compromised to ensure a fully functional utility to support FRC.

It is unlikely that all GDM design objectives will be met by FRC. This task will also result in a set of recommendations for phased release of GDM improvements in support of IPAD; it is presumed that these recommendations will result in follow-on studies, the costs of which are not included in this Baseline Implementation Plan but are considered as an optional extension plan in subsection 2.10.3.

2. **Manpower, Schedule, and Costs.** The nine-member team that participated in Blocks 67 and 68 will perform this task in 6 months, with a cost of 36 man-months and 30 computer hours.

2.5.7 Special purpose utilities (SPU). - Figure 2-12 shows the task sequence required to develop the three SPUs with a scheduled release 33 months after go-ahead and a cost of 117 man-months and 55 computer hours. There are three SPUs supporting IPAD:

1. **DML Insertion Preprocessor.** This is a batch preprocessor that scans the source code of existing FORTRAN programs, inserts the required FORTRAN DML source statements in place of the conventional FORTRAN I/O statements, assembles a preliminary set of FORTRAN SUBSCHEMA DDL, and assembles a set of diagnostics for those insertions that were uncertain or incomplete.
2. **SUBSCHEMA Assembler.** This is an interactive program that allows the programmer to examine the diagnostics resulting from the DML insertion phase and correct/complete the DML insertion, optionally compile the resulting FORTRAN program as a check, complete the SUBSCHEMA DDL as required, and complete the program-supporting IODEF, TCSs, and TCSSs.

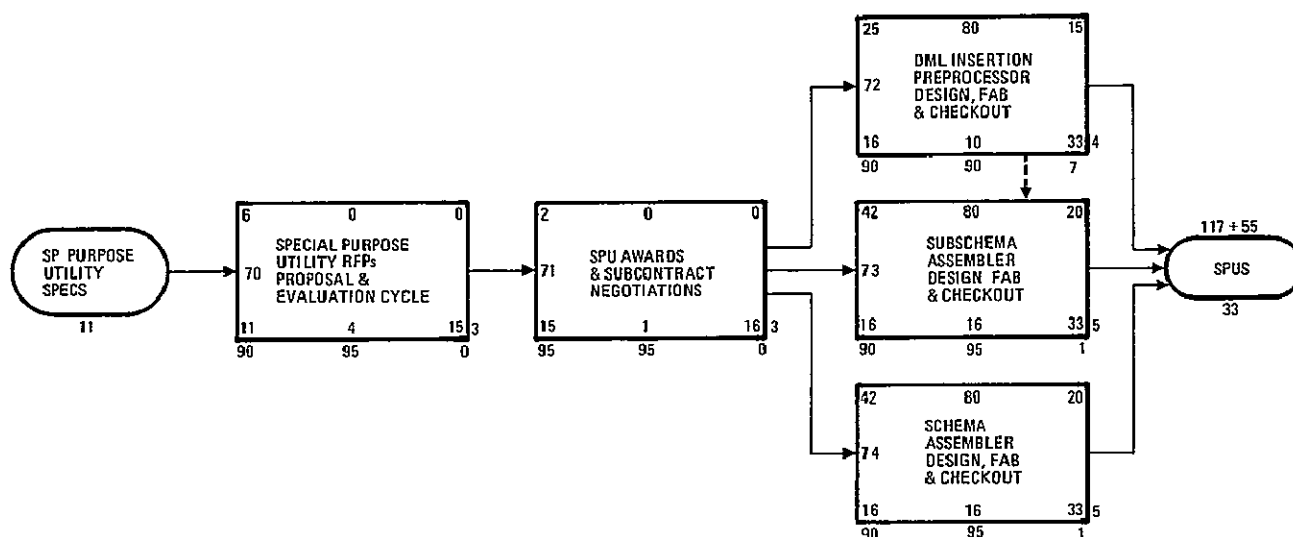


Figure 2-12. Development Phase — Special Purpose Utilities (SPUs)

3. **SCHEMA Assembler.** This is an interactive program that allows the IPAD user to assemble a SCHEMA to support the collection of programs (OMs and GPUs) supporting a specific task and to complete the SUBSCHEMAS supporting each program of that collection.

The Baseline Implementation Plan makes provisions to subcontract each of these utilities. Details of this plan are given in the following subsections.

ORIGINAL PAGE IS
OF POOR QUALITY

2.5.7.1 Block 70: The purpose of this task is to solicit and evaluate proposals for subcontracted development of each of the SPUs. The task will be performed as follows:

1. Approach. Numerous firms could bid on these tasks. Every major computer manufacturer employs such expertise in conjunction with their compiler development. These are several software houses that do this type of work, e.g., Computer Sciences Corporation (CSC) and LOGICON; indeed, some of these companies contract directly with the computer manufacturers for compiler tasks (e.g., CSC developed CDC's COBOL and JOVIAL). Further, several universities (e.g., Michigan State) might be interested in bidding the task. Correspondingly, there should be no lack of technically adequate proposals. Finally, since this is a very competitive field, the bids are likely to exhibit a relatively narrow price range.

Due to the expertise and the prerequisite code involved (e.g., general purpose lexical scanner and syntax analyzer), the contractor would absorb this task only as a last resort. There should be no reluctance, however, to split the interactive interface from the SUBSCHEMA Assembler and the SCHEMA Assembler — doing these inhouse — and subcontracting only the compiler-like code to an appropriate firm.

The proposal evaluation will be based on the bidder's understanding and appreciation of the problem and their management plan to ensure meeting schedules and costs with high product quality.

2. Manpower, Schedule, and Costs. A team of three members, including a data-base analyst, a system programmer/analyst, and a requirements analyst, will write the technical aspects of the RFPs and provide evaluations for the proposals submitted. Four months are estimated for this cycle, with a cost of 6 man-months.

2.5.7.2 Block 71: The purpose of this task is to award the SPU subcontract and conduct the pertinent negotiations. One month will be required for this task, with a cost of 2 man-months.

2.5.7.3 Block 72: The objective of this task is to design, fabricate, and check out the DML Insertion Preprocessor described in Volume V, Part I, Subsection 7.2.1. The task will be performed as follows.

1. Approach. The DML Inertion Preprocessor is a batch utility that preprocesses an input FORTRAN source program and inserts DML commands in replacing selective (or all) FORTRAN READ, WRITE, PRINT, or PUNCH statements. (Treatment of random files is also considered.) This utility

is essentially a Decision Table Processor that inserts the FORTRAN DML code and generates a cross reference table and incomplete DDL for the FORTRAN SUBSCHEMA. The resulting FORTRAN source program (with DML replacements) can then be compiled and executed using the DML-extended FORTRAN compiler or a standard FORTRAN compiler supported by a FORTRAN DML library.* This special utility is used one time per existing OM. In addition:

- a. The resulting code will not be nearly as efficient as a reprogramming of the OM with a complete reorganization of the input/output code and DML insertion by hand.
- b. It is unlikely that the preprocessor would be able to catch more than 80 to 95 percent of the revisions, and it may make incomplete insertions. Individual treatment must be accorded the rest by a DML-experienced programmer.

It follows that selective, heavily-used OMs should be subsequently reprogrammed (after insertion into IPAD by the preceding technique) at a convenient time.

This SPU is further complicated by the possible requirement to optionally replace conventional FORTRAN I/O with FORTRAN DML CALLs if the task group recommendation (Block 20) is for a FORTRAN language extension (rather than the conventional approach of extending FORTRAN via CALL statements). This is improbable, however; see the related discussion in subsection 2.4.6.3.

The approach to be taken is to allow the subcontractor to fabricate and check out the code after the design has been completed and approval obtained from the contractor. Extensive checkout of the code will be easily obtained by supplying existing FORTRAN source programs of varying I/O complexity for processing. The proficiency of the preprocessor — viz., the percentage of insertions made correctly (Item b) — can be evaluated at the same time.

*Which approach is used depends on the recommendation of the DML task group (Block 20) and on the time frame. If the group recommends a FORTRAN language extension and the FORTRAN temporary code is no longer required, this implies that the manufacturer has supported the FORTRAN DML with a FORTRAN compiler extension.

2. Manpower, Schedule, and Costs. The subcontractor is to supply a compiler programmer and a data-base analyst. The contractor will supply a data-base analyst for consultation and review, and selected personnel accompanying candidate OMs for testing DML insertion.

Since the most likely case is to supply FORTRAN DML via CALL statements, the cost estimates for Block 72 are for this case. If the FORTRAN DML were to be accomplished via FORTRAN source-code extension, the preprocessor must optionally be able to insert both the FORTRAN DML (as the long-term solution) and the equivalent FORTRAN DML CALLs (as the short-term solution detailed in Block 51). The added cost would be eight man-months and five computer hours.

Several factors are significant in reducing the cost of developing this code in comparison to developing typical compiler code, which could be some four to five times greater:

- a. The DML Insertion Preprocessor is working with source code and producing source code. There is no requirement to produce "efficient" object code as must a compiler. In fact, there is no requirement to produce object code at all.
- b. A compiler is expected to handle the specified language completely and to provide reasonably complete diagnostics when this cannot be done. The DML Insertion Preprocessor is expected to handle "most" cases (particularly the conventional ones) so as to relieve the burden on the programmer in inserting OMs. It is allowed to get stumped and quit on more difficult substitutions, and even to make incomplete substitutions in certain circumstances. The diagnostic requirements are merely that the programmer is clearly informed of the preprocessor's quandary when it gives up on an insertion.
- c. Item b is a grey area. The proficiency of the preprocessor can be adjusted "slightly" with a resulting large variation in costs. (This is typical of systems approaching perfection as an asymptote. Even the best compilers make occasional errors.) It is intended to use this fact to manage costs by compromising proficiency within reasonable limits.

This task is estimated to last for 10 months, with a cost of 25 man-months and 15 computer hours.

3. Risk Assessment. The risks of this task are due to the possibility of encountering problems in excess of those anticipated. Technically, this is a

relatively low-risk task to develop software that is very similar to the front end of an ANSI-standard FORTRAN compiler.

2.5.7.3 Block 73: The objective of this task is to design, fabricate, and check out the SUBSCHEMA Assembler described in Volume V, Part I, Subsection 7.2.2. The task consists of:

1. Approach. The SUBSCHEMA Assembler is an interactive utility with which an experienced programmer and/or user completes the DDL for the OM's SUBSCHEMA source using the incomplete DDL, the cross reference table, and the reworked FORTRAN source code produced by the DML insertion Preprocessor. Integral with this process is a renaming (by equivalence statements) of the variables involved to make them more meaningful from the user's viewpoint. In some selected instances, the programmer/user may recognize the need/desire for reorganization and end up by modifying the OM's FORTRAN source code as it relates to the variable's internal format. The resulting DDL for the OM's SUBSCHEMA is then complete and correct, as could be verified by the FORTRAN-extended DDL compiler. This is generally a one-time task per OM, although it is conceivable that subsequently desired FORTRAN modifications could originate with this utility.

As discussed in Volume V, Part I, Subsection 7.2.2, this utility may be combined with the DML Insertion Preprocessor, obviating the requirement for a separate preprocessing phase.

The approach to be taken is to allow the subcontractor to fabricate/check out the code after approval of the detailed design has been obtained from the contractor. It is envisioned that early extensive checkout can begin as required inputs become available from the DML Insertion Preprocessor (Block 72). Block 72 (which has substantial slack) can be scheduled to complete ahead of Block 73 so as to provide the required input.

2. Manpower, Schedule, and Costs. The subcontractor will supply a compiler programmer, a data-base analyst, and a programmer skilled in the design of interactive systems. The contractor will supply a data-base analyst and selected personnel as required for review of the suggested interactive interface. This task is scheduled for a duration of 16 months, with a cost of 42 man-months and 20 computer hours.
3. Risk Assessment. This is technically a low-risk task similar to interactive text editors, particularly those used for interactive programming.

2.5.7.4 Block 74: The objective of this task is to design, fabricate, and check out the SCHEMA Assembler, described in Volume V, Part I, Section 7.3. The task will be performed as follows.

1. Approach. The SCHEMA Assembler is (principally) an interactive utility (it must also be able to run batch) with which an inexperienced user assembles his User File (UF) SCHEMA source from a collection of SUBSCHEMAS (source), one for each OM he intends to use in the course of an assignment. Using this program, the user:
 - a. Reconciles the various names of the same variables in the SUBSCHEMAS and selects the desired name to represent this variable in the SCHEMA, renaming as required.
 - b. Selects the appropriate variable's format for the SCHEMA when there are conflicts among the variable's format in the various SUBSCHEMAS.
 - c. Constructs data base procedures (e.g., units conversion) as might be required.

This is a many-time task supporting the user in each assignment (study) he initiates. It isolates him from the DDL necessary in describing his UF SCHEMA and (optionally) allows him to recompile his UF SCHEMA and append it to the IPAD Project SCHEMA (as a modification to his AREA).

The approach to be taken is identical to that described in subsection 2.5.7.3 for Block 73, and the reader is referred to that subsection for a directly applicable discussion.

2. Manpower, Schedule, and Costs. The personnel skills, task duration, and costs are identical to those of Block 73.

2.6 Phase I - Task 3, IPAD First Release Capability System No. 1

The objective of this major task is to check out IPAD's subsystem and system software, culminating with a one-week demonstration of the First Release Capability (FRC) for computing system No. 1. Figure 2-13 shows that this major task is completed 45 months after go-ahead with an expenditure of 193 man-months and 177 computer hours. The Pre-release Capability (PRC) is available 33 months from go-ahead and is to be checked out with the combined participation of an engineering team to

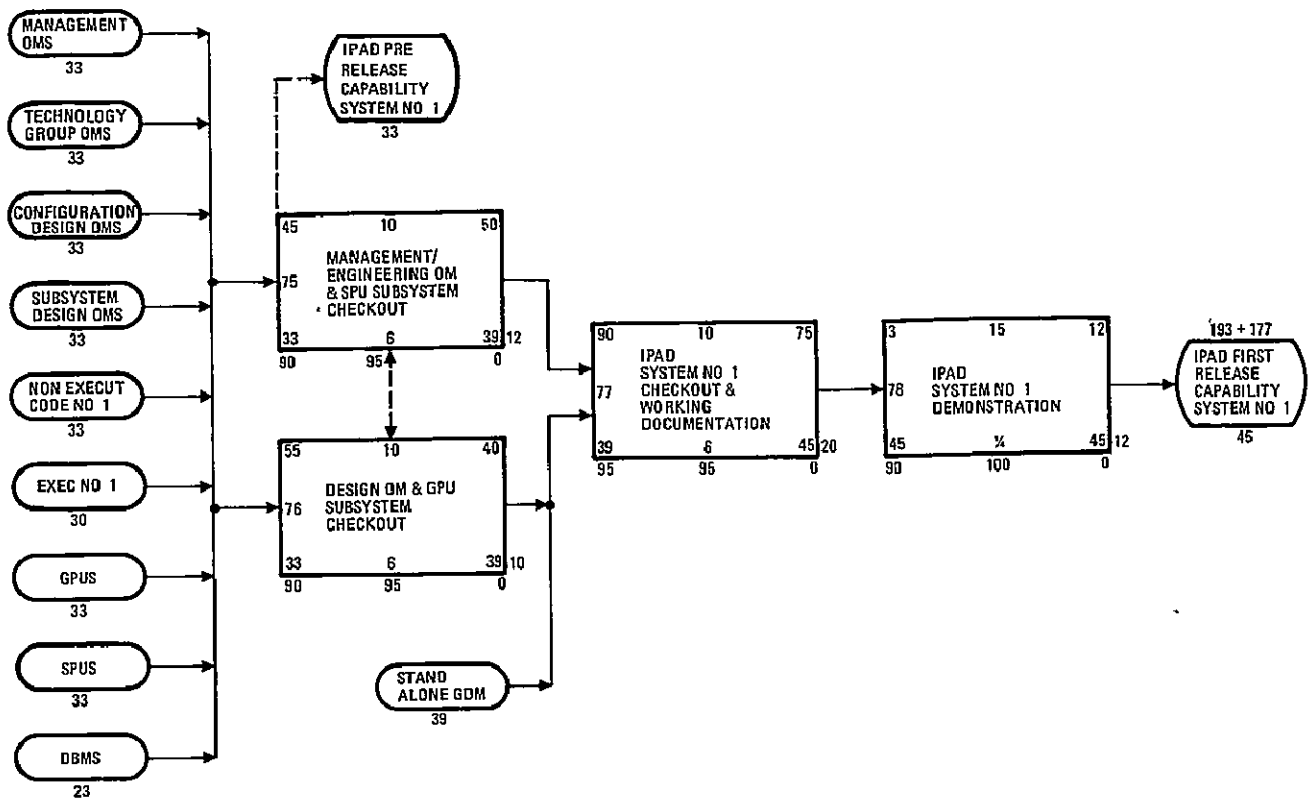


Figure 2-13. Phase I - Task 3, FRC Checkout and Demonstration

exercise the various software modules within a project design framework, and a team of computer system personnel to integrate and check out the systems software/hardware, to monitor operations, and to provide guidance in the use and troubleshooting of the various features of the IPAD System.

The engineering-user team is envisioned at this time to be composed of eleven members to fulfill the following functions:

1. Project Management, to provide technical direction as well as exercise the new interactive project management capability developed for IPAD.
2. Data Base Administration, to have responsibility for structuring, assembling and controlling the project-related multidisciplinary data bank.
3. Economic Analysis/Operations Research evaluations, to provide project support in these technology group areas.
4. Configuration Design, to exercise the GDM and the vehicle configuration design capability while supporting the project-oriented activity.

5. Aerodynamics/Performance/Flight Control evaluations, to generate the project data required from these disciplines while using the related OMs in the IPAD environment.
6. Propulsion System evaluations, to provide data applicable to the project and to perform studies with applicable OMs selected for IPAD demonstration.
7. Mass Properties studies, to assist in the generation of project data pertinent to all subsystems and in assembling the data bank.
8. Three subsystem designers, in the areas of structures, propulsion and fuel systems, and power systems, to exercise the GDM in support of design activities related to the project.
9. Loads/Structural Analysis/Dynamics/Materials studies to apply the pertinent OMs in support of project activities and properly interface these closely related technology group areas.

Although some overlap of functional groups is apparent in the team composition, it is assumed that each member will have part time assistance as required from specialists in each group.

The other team, composed of computer system personnel as detailed below, will parallel the efforts and assist the engineering team. The computer-system team will be responsible for making the PRC operational progressively and implementing changes and improvements deemed necessary during this checkout task.

The computer system team is envisioned at this time as composed of eleven members (all of them previously associated with IPAD development) to provide assistance in their respective fields of specialty. The proposed breakdown is:

1. Three System Programmer Analysts.
2. Three Senior Applications Programmers
3. Two Senior Graphics Programmers.
4. Two Data Base Analysts.
5. One Requirements Analyst.

Specifically, the engineering and computer-system checkout teams will be required to:

1. Uncover potential problems and find satisfactory solutions.
2. Improve on the code where necessary or where cost-effective.
3. School themselves (the future demonstration team) in:
 - a. The system in general.
 - b. Their subsystems in particular.
 - c. The exploitation of IPAD user features.
 - d. The available documentation.
4. Construct and improve product demonstrations.
5. Construct and run benchmarks intended to quantify the improvements to be attained through IPAD.
6. Construct and improve questionnaires designed to aid in product evaluation.
7. Anticipate likely future operational problems, define these and postulate likely solutions. (This will aid in their discovery and resolution during product demonstration).
8. Encourage pre-release production work by interested parties.
9. Upgrade the documentation.

All the software elements that heretofore have been developed or refurbished to provide the capability previously illustrated in Figure 2-2 will be systematically integrated and checked as logical software subsystem groups.

Details of the checkout plan for this major task are given in the following subsections.

2.6.1 Management/engineering OM and SPU subsystem checkout, Block 75. - The purpose of this task is to provide an extensive checkout of the SPUs principally for the insertion into IPAD of technology-group OMs, which are required for the management and engineering team project activity and final IPAD-FRC demonstration. The task consists of:

1. Approach. Although all the software elements defining the PRC have been individually checked out and integrated with immediately related software, they have not yet been integrated in the context of a project-oriented activity. The approach proposed to develop and check out this emerging working tool capability is to assign the engineering team (as detailed below) a project-oriented design activity, which is meant to exploit IPAD's PRC and provide engineering user feedback for system and code corrections and improvements. The engineering team will use the project data in Block 32 and start assembling the project-related multidisciplinary data bank in parallel with their software integration and checkout activities.

The main objective of the engineering team is to exercise the IPAD capability. The amalgamating factor is the project-related activity.

The computer-system team will provide extensive checkout of the code, rectify problems uncovered with the operations and use of the SPUs, and make minor improvements as deemed necessary.

This combined checkout activity is considered an absolute requirement due to the following reasons:

- a. It is a very rare occurrence when a code module does not exhibit difficulties after it has gone into production. These difficulties are extremely aggravating in prototype systems since the user is never certain — through lack of familiarity — if the problem is his or the code's.
 - b. System design flaws often do not become evident until the system is put together. It is particularly important that the tests to check the system are representative of the actual usage environment.
 - c. The best system is worthless unless it is used. Its adaption by potential users is heavily dependent upon the system's reputation as a practical tool, and to attain this end goal potential users should be involved in its development and checkout.
2. Manpower, Schedule, and Costs. This task is scheduled for a 6-month duration with estimated costs of 45 man-months and 50 computer hours.



2.6.2 Design OM and GPU subsystem checkout, Block 76. - The objective of this task is to provide an extensive checkout of the GPUs, in particular QP, GGP and GDM in relation to engineering/design tasks for the selected aerospace-vehicle project.

This task is very similar to that of Block 75, and is performed by the same two teams previously discussed and for the same general and specific purposes. Both tasks are closely interrelated in terms of technical goals, schedule, and costs.

2.6.3 IPAD system No. 1 checkout, Block 77. - The purpose of this task is a concentrated effort on exploiting all features of the IPAD system from the engineer/designer points of view and focusing on the generation of product definition data and the assembling of the project data bank. The final objective of this task is to lead the way to a thorough and systematic demonstration of IPAD's First Release Capability. The task will be performed as follows:

1. Approach. The engineering team will increase its involvement toward project-oriented and user-features demonstrations while the computer-systems team will progressively get more involved toward the preparation of IPAD system working documentation, particularly as it relates to computing system No. 1.
2. Manpower, Schedule, and Costs. This task is scheduled for 6 months and its cost is estimated at 90 man-months and 75 computer hours.

2.6.4 IPAD system No. 1 demonstration, Block 78. - The objective of this task is to make a thorough demonstration of IPAD's First Release Capability. The task will be performed as follows:

1. Approach. The engineering and computer system teams of Block 77 will provide in-vivo demonstrations of all features of the IPAD system using demonstration cases that have been prepared in Block 77. This demonstration will be performed on the computer installations that heretofore have been used by the major IPAD contractor for the development and checkout tasks. Individual members of both teams will participate in the demonstration of features with which they are most familiar. The demonstration will follow a project development sequence from the initial definition stages to the most detailed activities conducted during IPAD system checkout.
2. Manpower, Schedule, and Costs. This demonstration is scheduled for a full week of activity with a cost of 3 man-months and 12 hours of computer time.

2.7 Phase I - Implementation Plan Management

The purpose of this activity is to manage and coordinate Phase I of the IPAD Implementation Plan described in the previous sections of this chapter. Figure 2-14 shows the two blocks associated with this task, which result in a completion date of 50 months after go-ahead and an expenditure of 121 man-months and 15 computer hours.

2.7.1 Management and coordination, IPAD system No. 1. - The management of the IPAD implementation tasks will be initially performed according to current practices using existing computerized techniques, mostly in a batch mode. After the first year, the interactive MANAGE No. 1 capability will be progressively used to perform the management function of the plan itself in the areas of control and trend charts, work sampling, information storage and retrieval, network modeling simulation, and forecasting. Management tasks for which automated capability has not been included in MANAGE No. 1 will continue to be performed according to current practices. At a later date, when MANAGE No. 2 is available, the additional capability will also be used in performing management tasks for the study itself. This block includes technical coordination for all aspects of Phase I.

2.7.2 Coordination meetings. - The implementation plan includes major management and technical coordination meetings every six months between NASA and the major

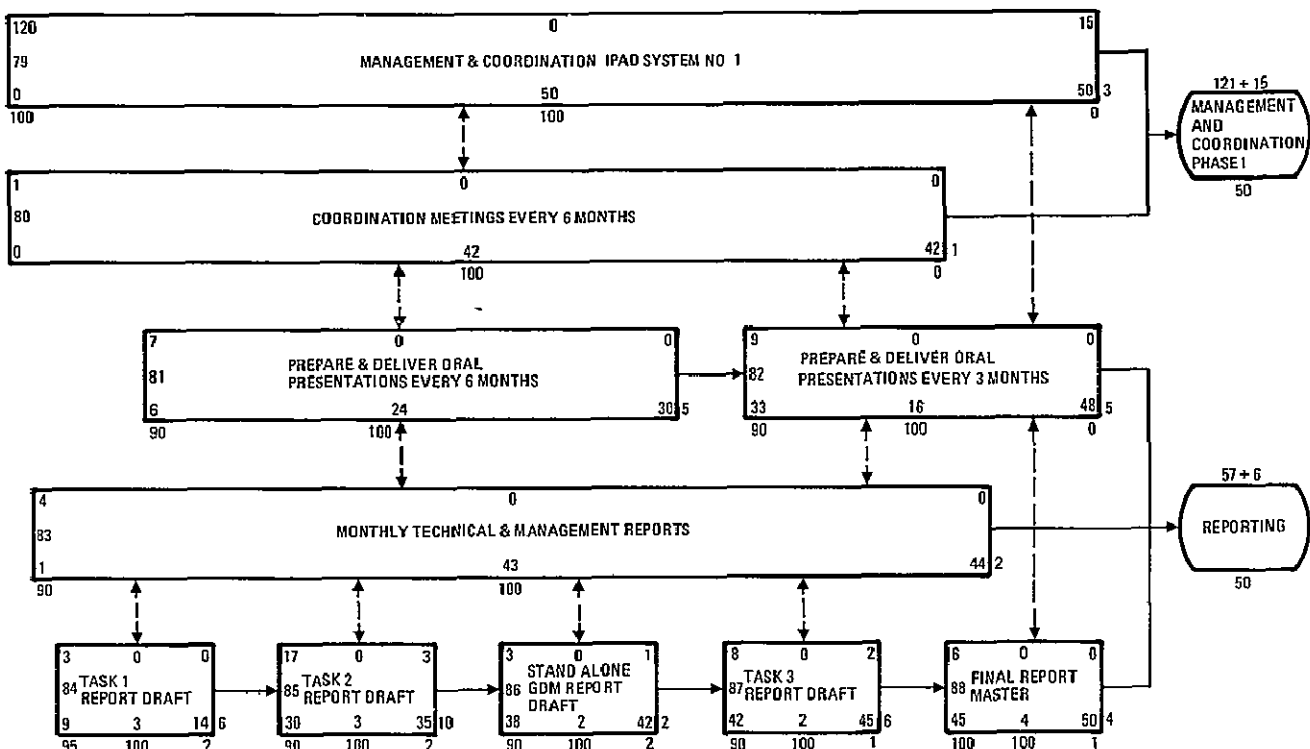


Figure 2-14. Phase I - Management and Reporting

IPAD contractor to review overall status of the plan, results achieved to date, and outstanding goals and milestones. The most recent developments from independent studies and computer technology advances will be evaluated periodically to determine their impact on the remainder of the implementation plan and to propose changes to it if worthwhile.

2.8 Reporting

The purpose of this task is to provide the study, technical, and management reports and presentations as well as the final documentation for each of the software elements and the IPAD User's Manual for system No. 1. This activity is completed 50 months from go-ahead at a cost of 57 man-months and 6 computer hours.

2.8.1 Oral presentations, Blocks 81 and 82. - The plan includes major oral presentations every six months during the first 30 months of the study and every three months thereafter. Full-day general presentations are scheduled, followed by pertinent working sessions for more detailed reporting to monitoring NASA personnel.

2.8.2 Monthly technical and management reports, Block 83. - This report will summarize all technical, management, and financial aspects of the study according to directives from NASA.

2.8.3 Final report, Blocks 84 to 88. - The final report will be divided into volumes associated with the major tasks and elements of IPAD and will be prepared according to the schedules and estimated costs shown in Figure 2-14. Separate documents will be prepared for engineering users and for detailed software documentation.

2.9 Phase II - Deployment to Computing Systems No. 2 and No. 3

This phase presents a separate implementation plan for two other computing systems, here termed systems No. 2 and No. 3 as illustrated in Figure 2-3. The selection of these two other systems was effected in Block 43 of Phase I (reference Figure 2-9, and subsection 2.5.3.1). The segregation of schedules and costs offered in this manner permits a clear view of the implementation costs for one system (including many common and transferable items) and the additional cost of deploying IPAD to other computing systems. Phase II is considered an integral part of the total IPAD implementation plan to make it available in major computing systems throughout the aerospace industry and governmental agencies. Further deployment to other computing systems can be effected in similar fashion but are not included in the Baseline Implementation Plan offered herein.

The basic approach taken for Phase II is to give participation to two other aerospace companies as major subcontractors to implement and check out the IPAD system

in their respective computing system installations. These installations must meet minimum hardware/software requirements as established in Block 9 of Phase I (reference Figure 2-6 and subsection 2.4.3.3).

Phase II is divided into main contractor tasks and subcontractor tasks for system No. 2 and system No. 3 as detailed in the following subsections. Phase II is completed 60 months after go-ahead at a cost of 642 man-months and 404 computer hours.

2.9.1 Main contractor tasks. - Figure 2-15 shows the sequence of tasks to be completed 60 months from go-ahead at a cost of 50 man-months and 6 computer hours.

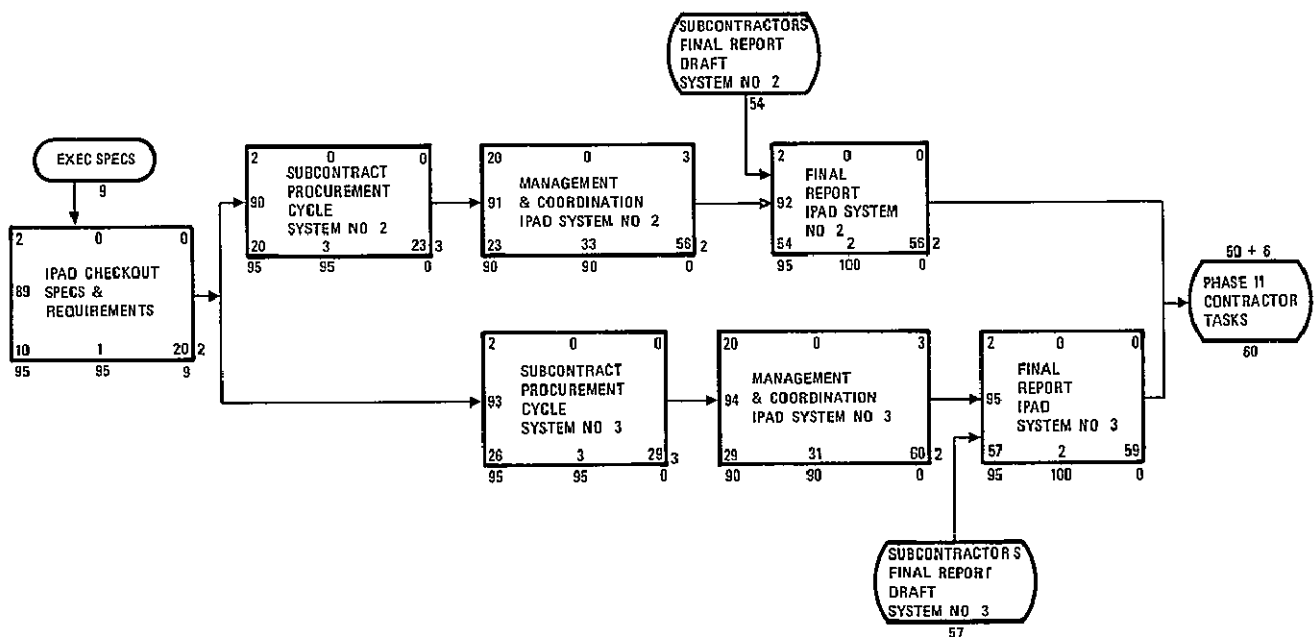


Figure 2-15. Phase II, Main Contractor Tasks

2.9.1.1 Block 89: The objective of this task is to provide a set of specifications for the minimum computing system required for checkout of the EXEC and subsystems of IPAD, and the additional computing equipment and project work required to demonstrate an IPAD project "in situ". The task consists of:

1. **Approach.** Compile a set of specifications encompassing the EXEC subsystem and system checkout/demonstration phases, to be used in the subcontract procurement cycle. These specifications will carefully outline the minimum equipment the subcontractor will be responsible for supplying, and the specific responsibilities of the subcontractor and contractor in the checkout/demonstration tasks. The results of Block 9 from Phase I will be used in the performance of this task.

2. Manpower, Schedule, and Costs. A team of two people will complete these specifications in one month at a cost of two man-months. The large slack time available will be used to delay the performance of this task so that advantage is taken of the findings of other Phase I tasks.

2.9.1.2 Block 90: The purposes of this cycle are to prepare a request for proposal to implement IPAD on system No. 2, to evaluate the responses, and to negotiate a subcontract with an aerospace company. The task consists of:

1. Approach. This procurement activity will be closely coordinated with NASA management and technical personnel, who will participate in reviewing the specifications (reference Block 89), in performing the evaluation of responses, and in the selection of the successful bidder.
2. Manpower, Schedule, and Costs. A team of three people will be involved in this cycle for 3 months and an expenditure of 2 man-months.

2.9.1.3 Block 91: The objective of this task is principally to provide technical coordination between the development studies of Phase I and the subcontracted studies. In addition, overall IPAD program management and integration will be provided for NASA. The task will be performed as follows:

1. Approach. The management interactive capability provided by MANAGE No. 1 (at month 12) will be initially used in the performance of this task, followed by the use of MANAGE No. 2 after its release (at month 30). The subcontractor will be offered this capability for conversion to his computing facilities.
2. Manpower, Schedule and Costs. A half-time involvement of a technical coordinator/manager will be required for a period of 33 months and an expenditure of 20 man-months and 3 computer hours.

2.9.1.4 Block 92: The purpose of this task is to review the final report draft prepared by the subcontractor and submit the final masters for NASA approval. This task will be spread over a period of 2 months at a cost of 2 man-months.

2.9.1.5 Blocks 93, 94, and 95: These tasks are identical to those of Blocks 90, 91, and 92 but with reference to a similar subcontracted effort to implement IPAD on system No. 3. Similar schedule and costs are assigned at this time, although it is recognized that many reasons could exist to make system No. 3 implementation different. Further considerations are considered too speculative at this time.

2.9.2 IPAD system No. 2, implementation plan. - Figures 2-16, 2-17, and 2-18 show the task sequence that will result in IPAD being deployed to computing system No. 2 fifty-one months after go-ahead and at a cost of 321 man-months and 202 computer hours (including one-half the cost of the contractor tasks).

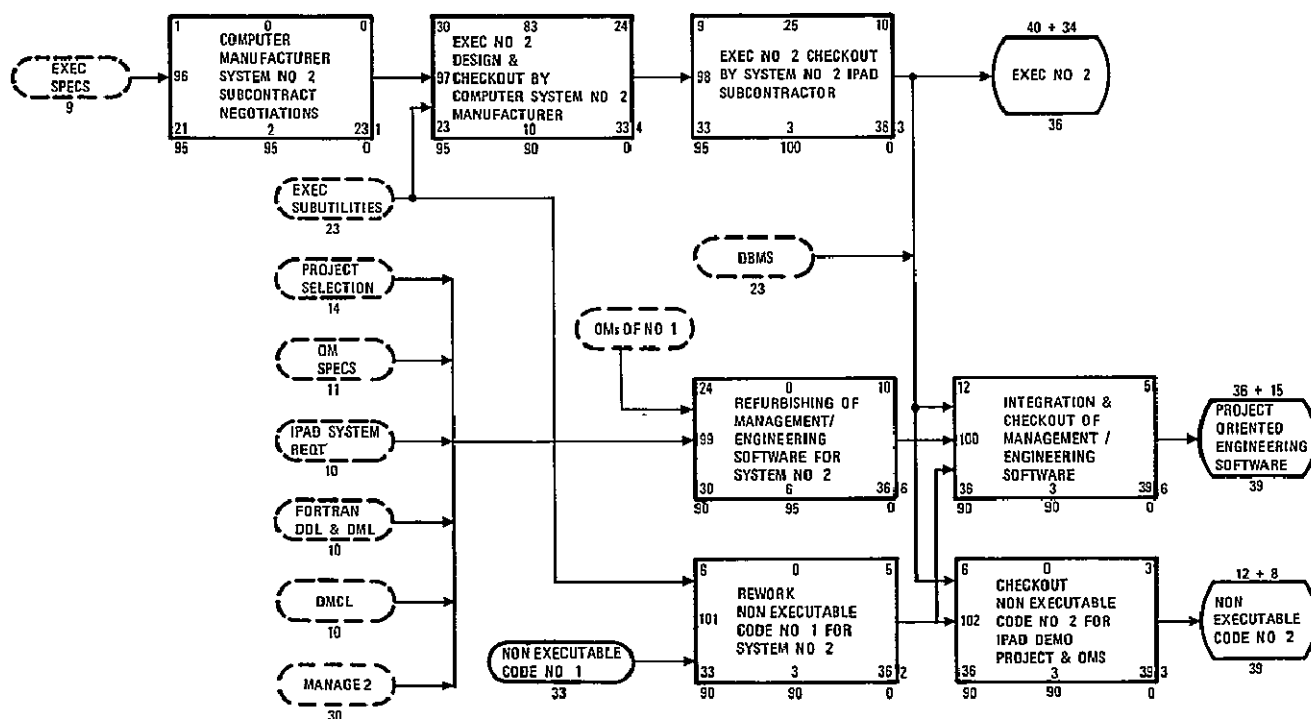


Figure 2-16. Phase II -- IPAD System No. 2 Implementation Plan (Identical for System No. 3 Except for Dates)

ORIGINAL PAGE IS
OF POOR QUALITY

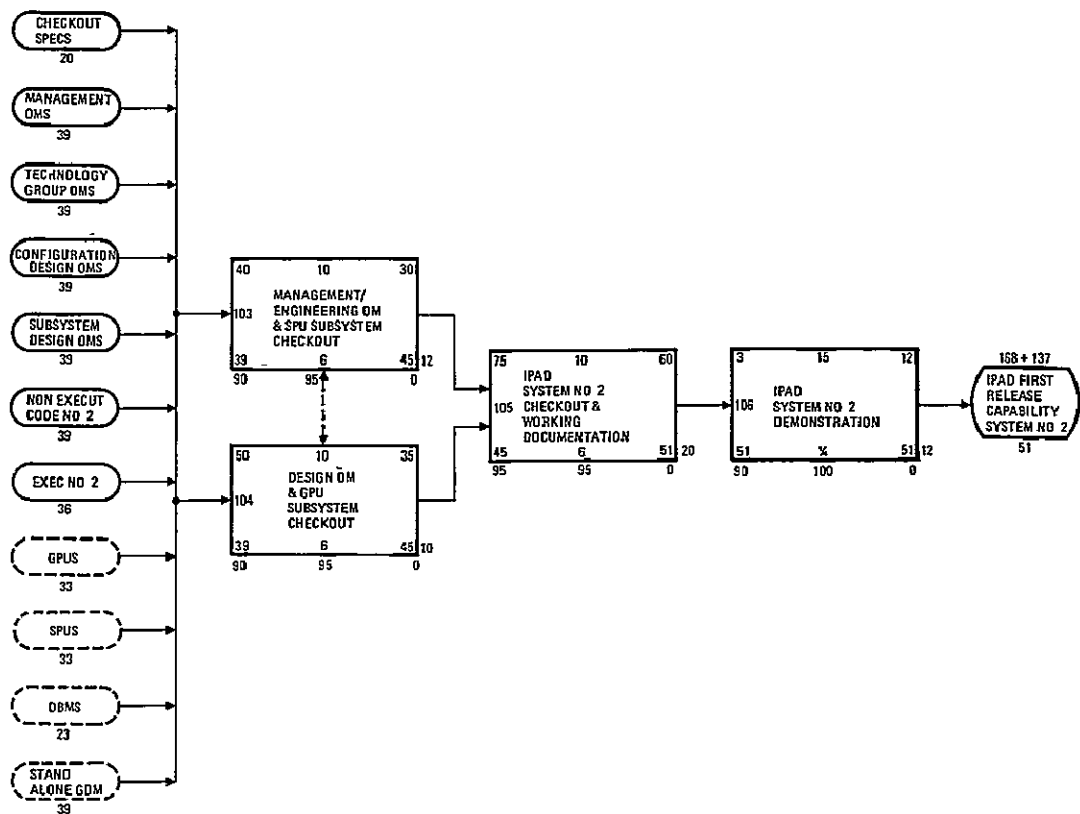


Figure 2-17. IPAD Implementation Plan, Phase II - Checkout and Demonstration System No. 2 (Identical for System No. 3 Except for Dates)

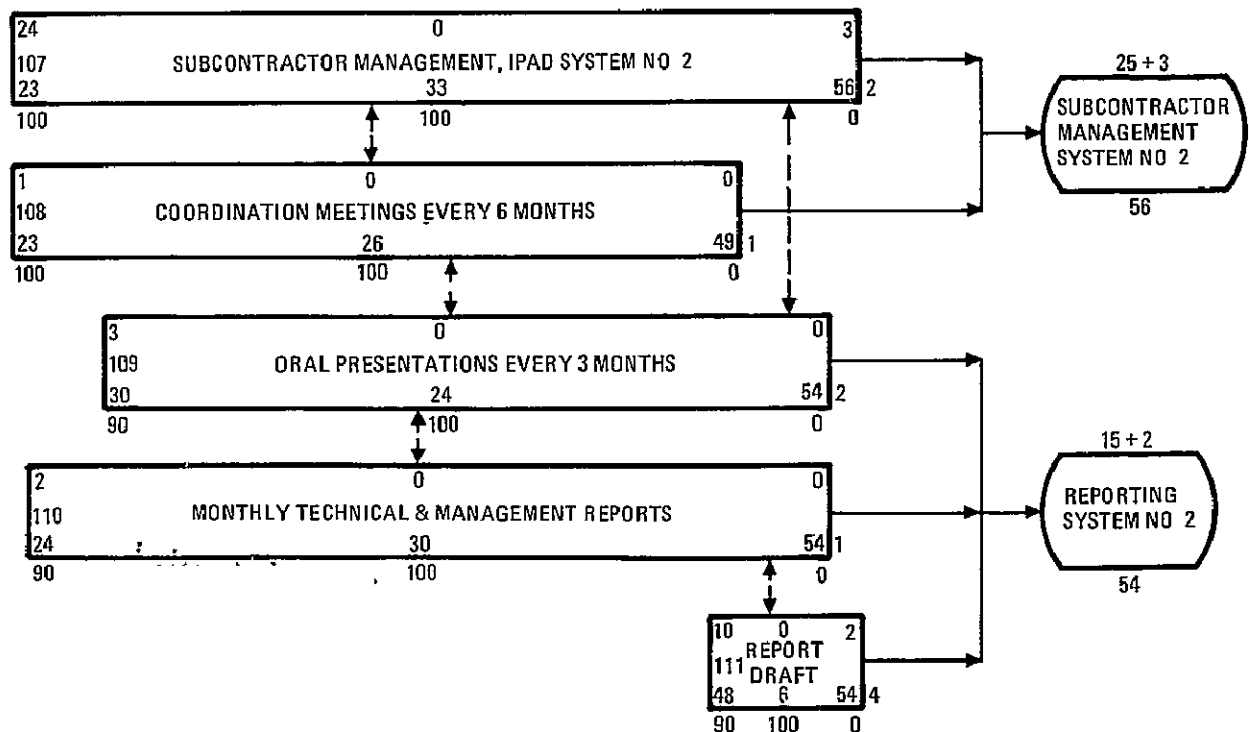


Figure 2-18. Phase II - Management and Reporting, System No. 2 (Typical for System No. 3)

All the development, checkout, and demonstration tasks will be performed by the subcontractor in his own computer installation and plant site. Technical coordination with the contractor will be mainly for availability and interpretation of data and working documents from Phase I concurrent studies. Details on the various tasks are given in the following subsections.

2.9.2.1 EXEC No. 2 development and checkout: The sequence of Blocks 96 to 98, Figure 2-16, yield the EXEC No. 2 program 36 months after go-ahead and at a cost of 40 man-months and 34 computer hours. These tasks are very similar to the sequence previously shown in Figure 2-9, for EXEC No. 1, except that the EXEC subutilities are immediately available from Phase I for the computer manufacturer's subcontracted task shown in Block 97. Due to their basic support of subsidiary processes and commands files, no modification to the host operating system is required. Thus, the design of IPAD's EXEC reduces to the best way to interface the EXEC's code with the host operating system. This is a straightforward task with few if any complications. For the sake of avoiding duplication of effort, the reader is referred to subsection 2.5.3 for a pertinent discussion of similar tasks.

2.9.2.2 Project-oriented engineering software: Blocks 99 and 100 summarize the tasks of refurbishing and checking out the management and engineering software that will be required for the project activity later on. These two tasks are very similar to those shown in the upper portions of Figure 2-8 (Blocks 33-39) and have identical objectives, except that they are for computing system No. 2. It is assumed herein that the general-purpose OMs widely used in industry (e.g., NASTRAN) will require little additional refurbishing for system No. 2 and can be readily adapted. On the other hand, it is fundamental to the success of IPAD that any engineering team be able to mount its own OMs, and therefore, it is proposed that the aerospace subcontractor for system No. 2 exercise its own choice of OM for IPAD demonstration, with the only constraint that the same aerospace-vehicle project (selected in Block 32) be used. This latter condition will simplify the checkout tasks, reduce costs, and permit a direct comparison of the results obtained with systems Nos. 1 and 2 (and 3).

Here again, the reader is referred back to subsection 2.5.2 for a directly applicable discussion of objectives and approaches to be followed for these tasks. The estimated costs are lower than for system No. 1, since many general-purpose OMs already refurbished for system No. 1 will require little additional effort for conversion to system No. 2. A completion date 39 months after go-ahead is scheduled for this tasks at a cost of 36 man-months and 15 computer hours.

2.9.2.3 Non-executable code No. 2: Blocks 101 and 102 give the required sequence of tasks that yield this code 39 months after go-ahead and at a cost of 12 man-months and 8 computer hours.

Since the non-executable code is somewhat computing-system dependent (e.g., word size, and the OSCL inbedded in the TCSs), it is intended to rework the code and repeat the task of Blocks 41 and 42 for computing system No. 2. The reader is referred to subsection 2.5.2.9 for a discussion directly applicable to the tasks for system No. 2.

2.9.2.4 IPAD first release capability, system No. 2: Figure 2-17 shows the sequence of tasks (Blocks 103 to 106) required to check out and demonstrate an FRC for computing system No. 2. The FRC is demonstrated 51 months after go-ahead at a cost of 168 man-months and 137 computer hours. This sequence is identical in objectives and approaches to that shown in Figure 2-13 for system No. 1. The reader is referred to Section 2.6 for applicable discussions on the subsystem and system checkout tasks and the demonstration of the FRC, system No. 2.

2.9.2.5 Subcontractor management: Figure 2-18 shows the block diagram for this task, which is completed 56 months from go-ahead at a cost of 25 man-months and 3 computer hours. The subcontractor will not be required to use the interactive MANAGE NO. 1 capability developed in Phase I, but it will be made available to him. The subcontractor will participate in coordination meetings between NASA and the contractor for discussion relative to his subcontracted tasks.

2.9.2.6 Reporting. Figure 2-18 shows the reporting tasks being completed 54 months after go-ahead at a cost of 15 man-months and 2 computer hours. These tasks include oral presentations to NASA scheduled on the same dates as the contractor's; monthly technical, management, and financial reports; and the preparation of a draft of the final report for IPAD documentation relative to system No. 2 and avoiding duplication of common items already reported in Phase I.

2.9.3 IPAD system No. 3 implementation plan. - This implementation plan is almost identical to that for system No. 2 except for being phased 6 months behind, and small differences in reporting schedules toward the end of the study. The reader is referred to Figure 2-3 for an overview, schedules, and costs associated with these tasks and to subsection 2.9.2 for applicable discussions.

2.10 Alternative Approaches

Three developmental alternatives were identified in conjunction with the base-line plan.

1. Form an "ad hoc" task group to develop the required GGL language interface (discussed in subsection 2.4.4.1).
2. Form an "ad hoc" task group to develop a recommended FORTRAN DML and SUBSCHEMA (and SCHEMA ?) DDL (discussed in subsection 2.4.6.2).

3. Provide an improved version of GDM to be available following FRC (discussed in subsection 2.5.6.12).

These alternative subplans are discussed in subsections 2.10.1 through 2.10.3; all result in increased costs. Subsection 2.10.4 discusses several alternatives designed to reduce cost of the baseline plan through compromise of the design objectives.

2.10.1 GGL language development. - This task would replace Blocks 10 and 11 (see Figure 2-6) as noted in Figure 2-19.

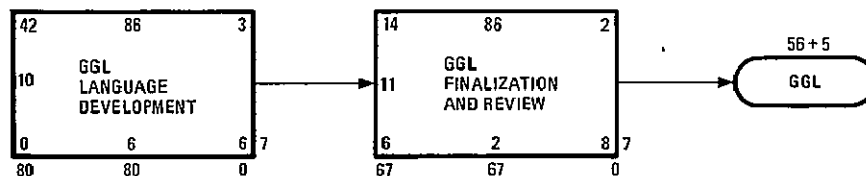


Figure 2-19. Alternative GGL Development

Its objective is to investigate and specify requirements for IPAD's GGL with the intent of eventually getting GGL adopted as a national standard. The tasks shown in Blocks 10 and 11 of the sketch above have the following characteristics:

1. Approach. It is assumed that the required language development would be sponsored by NASA to ensure meeting IPAD schedules. The required task group would consist of six representatives from industry, a chairman, and representatives invited from the various hardware manufacturers. The industry representatives and the chairman would be full time, under subcontract to IPAD's contractor. The contractor would additionally underwrite publication expenses. Since the intent is to develop a language specification that could eventually become a national standard, it would be acknowledged that the contractor would exert no technical control over the sponsored task group; neither would the members be held accountable to their companies for their recommendations.

It is anticipated that sponsored-member selection would be accomplished through RFP for the individual's services, source selection being determined by qualifications and interviews (principally by the selected chairman).

The task group is to meet monthly for a period not to exceed five days at a place selected on rotation to minimize travel. Each sponsored member would return to his company with tasks to prepare for their next meeting.

The GGL Task Group will be formed for the expressed purpose of developing the graphics interface specifications for IPAD applications. This task will begin with a comprehensive review of existing graphic libraries and the syntax of their CALL statements. The required primitives will be functionally specified as well as desirable higher-order CALLs to enable minimization of required code for typical applications.

Considerable effort will be expended in deriving the desired syntax of the FORTRAN CALLs, paying particular attention to defaulting to the corresponding DVST capability when not on a refreshed device. The desired subroutine structure and argument arrangement will then be evaluated in the light of existing (and planned) capability to assess the impact of the tentative syntax on existing (or soon to exist) software. Review of the syntax specification will be considered at this juncture.

Once the syntax specification is complete, the recommended lexicon for the subroutine names and the arguments will be developed with the objective to enhance usability.

2. Manpower, Schedule, and Costs. The industry participants must all be graphic programmers with extensive experience on their graphics systems. It is desired that they be engineer/programmers. At least one programmer representing each of the target computing systems and a representative of the DVST applications must be in the task group. The selection of a chairman is critical to the effective operation of the task group. An 86 percent subcontract factor provides the contractor with one member in the task group.

A preliminary specification for GGL (combining and extending the best features of the available languages) should result from Block 10. Two additional months are presumed for review, correction and approval of the specification. (Publication of the specification is to follow Block 11.) Three host computer hours are sufficient to prepare the preliminary specification using a computer featuring an interactive text editor with an upper and lower case character set. Two additional hours should suffice for review copies and corrections. By comparing the sketch above with Figure 2-6, it can be seen that this alternative plan increases the required first-year spending by 51 man-months and 2 computer hours.

3. Risk Assessment. The critical period where slippage could occur is during task startup. It is here that the task group chairman is most instrumental in assisting the group in partitioning tasks and setting up individual responsibilities. A technical risk associated with Block 10 is that the task is more difficult than anticipated. An 80 percent confidence was estimated for costs and schedule in Block 10.

The principal risk associated with Block 11 is that still unresolved technical issues may preclude achieving consensus. Here again, the task group chairman must act as the arbitrator to see that compromise is obtained without jeopardizing the main objectives. A confidence factor of 67 percent was assessed for both costs and schedules in Block 11.

4. Alternatives. No alternative has been identified in case of larger costs or schedule slippage, other than trying to absorb them in subsequent related tasks.

2.10.2 FORTRAN DML and DDL language development. - This task would replace Block 17, 19, and 20 (reference Figure 2-6) as noted in Figure 2-20.

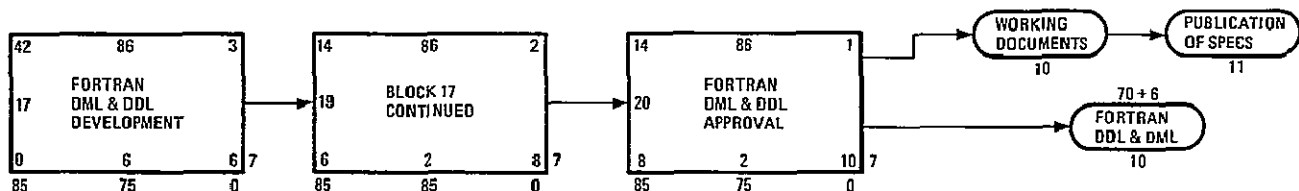


Figure 2-20. Alternative DML and DDL Language Development

Its objectives are to investigate and specify requirements for a FORTRAN DML, a FORTRAN SUBSCHEMA DDL and (if recommended) a FORTRAN SCHEMA DDL to interface with CODASYL's DBMS. These replacement task blocks have the following characteristics:

1. Approach. The nature and construction of the task group is identical to that discussed in subsection 2.10.1. This is the same task as discussed in subsections 2.4.6.2 through 2.4.6.6, with the exception that
 - a. It is assumed that the task group does not have CODASYL's backing (at least not initially).

- b. The group will be composed of seven rather than 25 members.
 - c. The resulting specifications and subsequent review will not be as comprehensive as those envisioned for the CODASYL-sponsored task group.
2. Manpower, Schedule, and Costs. The task group members must be data base analysts with multi-language experience and a solid background in FORTRAN. At least one member must represent each of the target computer systems. This is essentially a ten-month task for the seven-man task group, resulting in a cost of 70 man-months and 6 computer hours. Blocks 17 and 19 represent a split block for connectivity. Correspondingly, the development phase takes eight months, and review and approval an additional two months. (Publication of the specification is to follow Block 20.) The 86 percent subcontract factor provides the contractor with one member in the task group. By comparing the sketch above with Figure 2-6, it can be seen that this alternative plan increases the required first-year spending by 65 man-months and 5 computer hours.

2.10.3 GDM improvements. - Figure 2-21 shows the sequence of task blocks that will permit improvements in GDM, making use of new software modules and feedback from the design team activities of the checkout phases for all three computing systems.

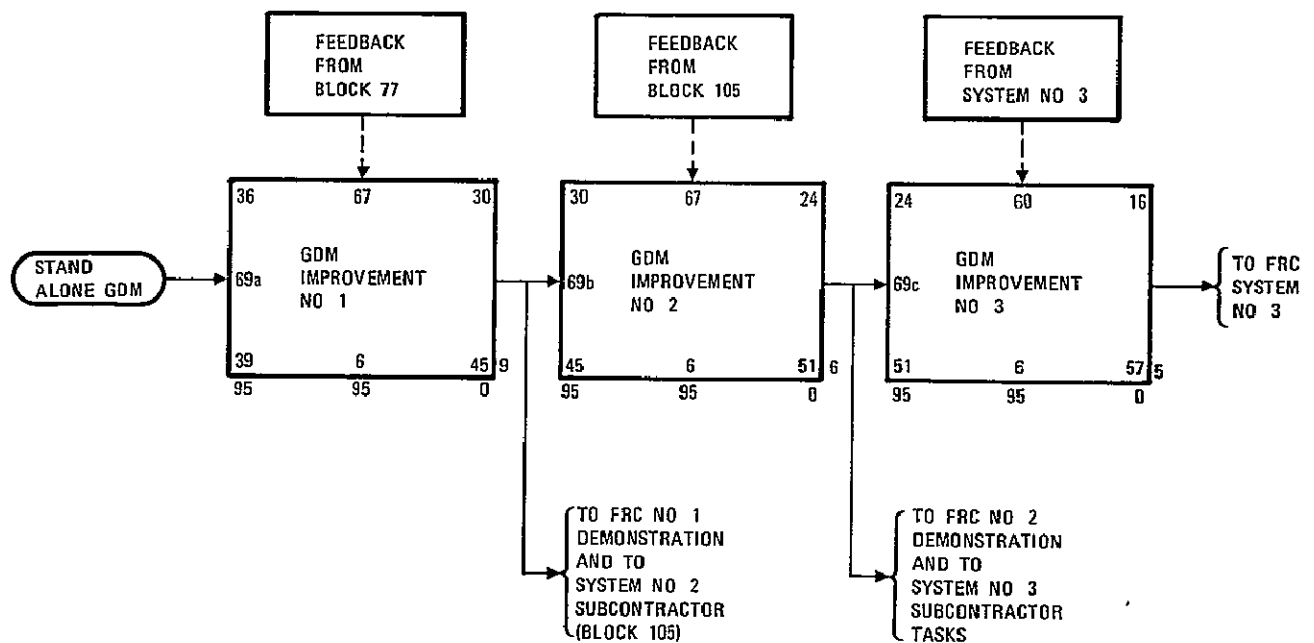


Figure 2-21. GDM Improvement Plan

The objective is to continue the development of GDM to the point where all design objectives have been met, and the recommended improvements have all been incorporated. This task was discussed in subsection 2.5.6.12, and the optional extension plan proposed herein is an extension of Block 69 to incorporate, principally, improvements related to engineering-usage features. The characteristics of these tasks are:

1. Approach. The team assigned to these tasks is the same as for Block 69 initially and is progressively decreased in size as less improvements or corrective actions are expected as the job moves from one block to the next. The designers from the engineering team of Block 77 will provide user feedback to the GDM improvement team, who in turn will separately implement those changes deemed worthwhile or necessary. The improvements introduced in Block 69a will be used in FRC No. 1 and will be made available to the aerospace subcontractor of system No. 2 for the IPAD subsystem check-out task of Block 105.

Blocks 69b and 69c have the objectives of continuing the GDM improvements while focusing on the needs and feedback from the engineering teams of the aerospace subcontractors for systems No. 2 and No. 3.

2. Manpower, Schedule, and Costs. The GDM improvement team will take 18 months to complete at a total expenditure of 90 man-months and 70 computer hours.

2.11 Cost Distribution Curves

This section presents distribution curves for both man-months and computer hours. Figure 2-22 presents these distributions broken down for each of the major computing systems included in the baseline plan, and paralleling the plan overview shown in Figure 2-3. It can be seen that for development of IPAD in one computing system, the larger expenditure rates occur between 24 and 34 months from go-ahead with peaks of 45 man-months per month and 25 computer hours per month.

The cost distributions for the development of IPAD for system No. 2 are given by the second set of curves, which show the larger rates between 38 and 48 months from go-ahead with peaks of 16.5 man-months and 12.5 computer hours per month. For system No. 3, the curves are almost identical to those for system No. 2 but phased about 6 months behind.

The total distribution curve is shown in the lower set of curves, indicating the larger rates of expenditure between 24 and 34 months from go-ahead, with peak rates of 58.5 man-months and 33.5 computer hours per month.

Table 2-1 gives the breakdown per fiscal year and per computing system for the Baseline Implementation Plan total costs, assuming a go-ahead date at the start of FY 1975.

TABLE 2-1. IPAD BASELINE IMPLEMENTATION PLAN, COSTS PER FISCAL YEAR

Fiscal Year	System No. 1		System No. 2		System No. 3		Totals	
	Man-Months	Computer Hours	Man-Months	Computer Hours	Man-Months	Computer Hours	Man-Months	Computer Hours
1975	220	51	-	-	-	-	220	51
1976	444	180	12	4	-	-	456	184
1977	495	279	84	50	40	16	619	345
1978	210	159	160	103	129	76	499	338
1979	16	1	65	45	152	110	233	156
Totals	1385	670	321	202	321	202	2027	1074

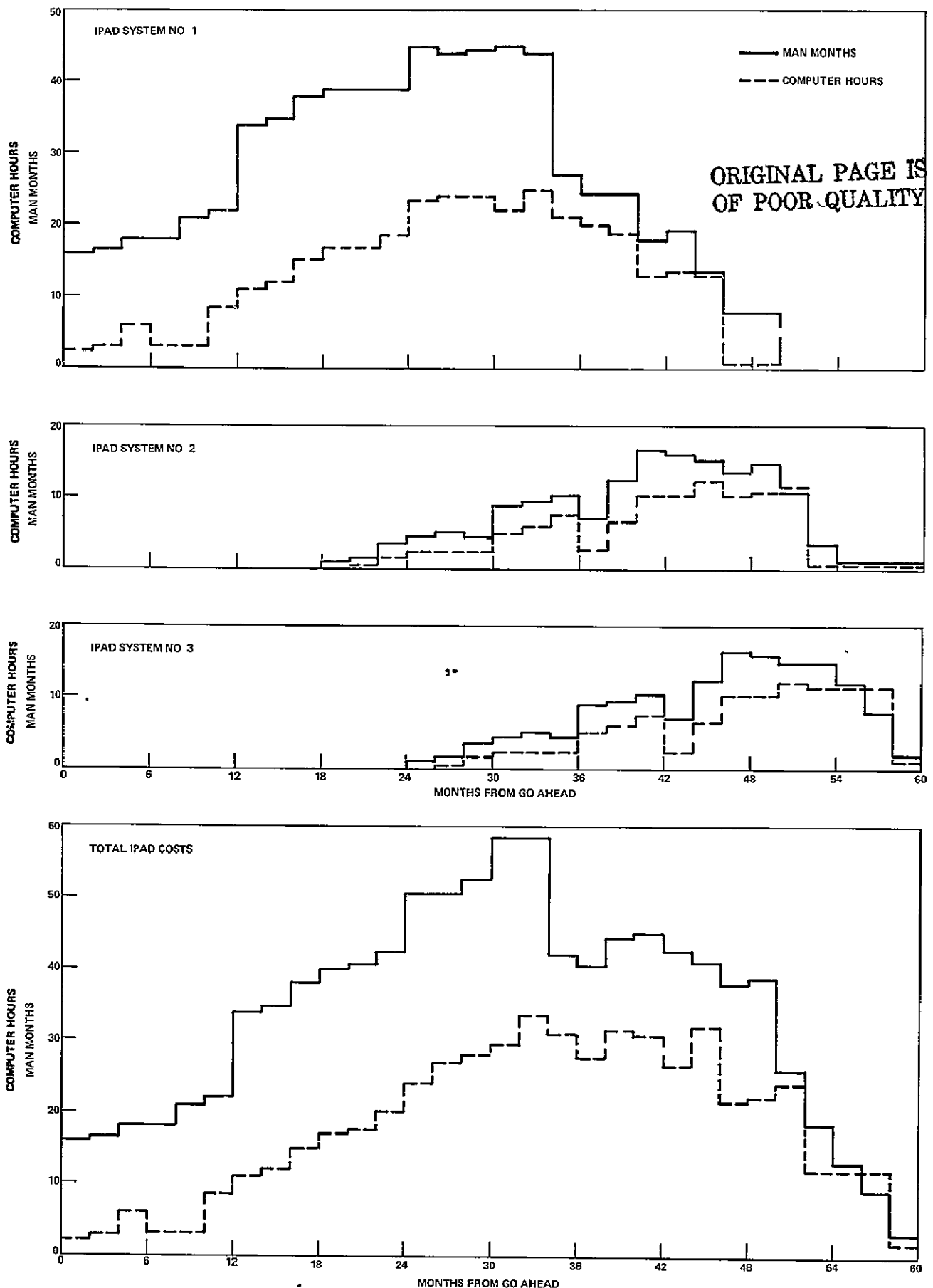


Figure 2-22. Cost Distribution Curves for Man-Months and Computer Hours

2.12 Variants to Reduce Peak Funding and Costs

This section presents the results of a quantitative evaluation to reduce peak funding and a qualitative assessment of various alternatives to reduce program costs.

2.12.1 Variant to reduce peak funding. A variant to the baseline plan was investigated with the objective of reducing peak expenditure rates. The man-month distribution curves obtained for a 6-month selective stretchout are shown in Figure 2-23 for all three systems. The curve of total cost indicates that the rate of expenditure can be maintained below 40 man-months per month, which is a reduction of 34 percent from the peak expenditure rate of the baseline plan. The breakdown of man-months per fiscal year is given in Table 2-2 for the total costs. A slight increase in cost is due to additional coordination.

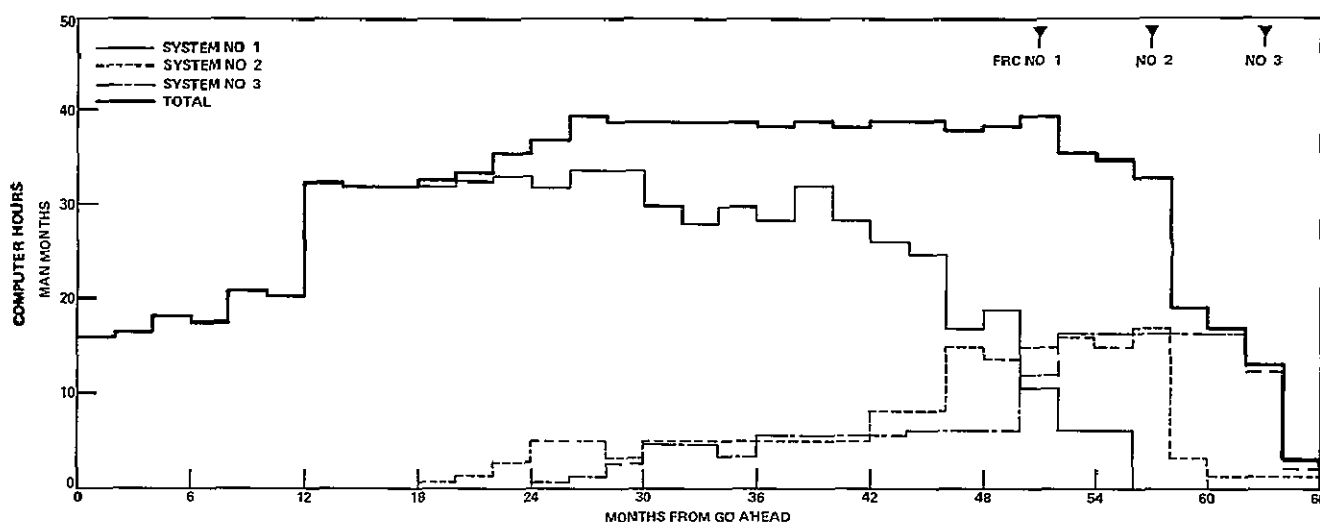


Figure 2-23. Six-Month Stretchout Variant to IPAD Implementation Plan

TABLE 2-2. 6-MONTH STRETCHOUT IPAD IMPLEMENTATION PLAN BREAKDOWN PER FISCAL YEAR

Fiscal Year	Man-months
1975	220
1976	398
1977	470
1978	470
1979	410
1980	80
	2038

2.12.2 Alternatives designed to reduce costs. The purpose of this subsection is to explore methods to reduce the peak funding and/or costs of the overall plan. Table 2-3 presents the more realistic alternatives which serve to reduce costs.

TABLE 2-3. IPAD DEVELOPMENT ALTERNATIVES TO REDUCE COSTS

	<u>Advantages</u>	<u>Disadvantages</u>
Delay support from manufacturers #2 and #3	a. Reduces peak funding	a. Delays IPAD on other computers b. GPU RFPs restricted to one computing system's users. c. Reduction in "competitive incentive" to manufacturers and GPU bidders d. All eggs in one basket
Circumvent GGL	a. Reduces cost slightly (no GGL specs required) b. Reduces peak funding	a. GPUs not transferable, must be re-programmed for computers #2, #3. b. Subsequently developed graphics OMs and GPUs not transferable. c. Much needed national standard delayed substantially
Delay STATUM	a. Reduces peak funding slightly	a. Insignificant peak-funding reduction b. Useful utility not available at first release.
Delay GDM	a. Reduces peak funding substantially b. Significant schedule relaxation. c. Increases competition d. Probably reduces cost	a. Principal GPU used by designers not available. b. Longest learning cycle code comes last. c. Designers cannot support IPAD-assisted project schedules d. Higher risk to IPAD success
Do not provide training and demonstration	a. Less cost b. Earlier contract termination	a. Delay acceptance of IPAD b. Delay benefits provided by IPAD c. Higher risk to IPAD success

It must be recognized that the baseline plan was a minimum-risk, minimum-cost plan; none of these alternatives can be recommended.

ORIGINAL PAGE IS
OF POOR QUALITY

3 IPAD SYSTEM OPERATIONAL COSTS (PHASE II, TASK 5)

Operational costs as required to support an IPAD environment are concerned with computing-facility hardware, computer support and maintenance, software maintenance, and the use of time and human resources.

The computing-facility hardware, support, and maintenance costs include main computers, minicomputers, communications interface, various types of terminals, computer operations, system software maintenance, and accounting support. These costs are typically combined into a computer allocation charge that is passed on to the user in the form of a rate per usage unit. This is usually a direct charge; however, the cost distribution algorithm and whether any of the costs are allocated to overhead can have a significant effect on the operational costs to a project.

The cost distribution algorithm strives for two things: (1) ensure that the charges against the projected computer work load will pay the computer hardware and support costs, and (2) try to distribute the user charge equitably across the hardware used. For an IPAD environment, it appears that the best way to handle interactive terminal charges is to have an easily visible "residency" charge that is separate from the actual computer computation cost. This approach lends itself to more accurate comparison of costs in doing jobs interactively when the former method may or may not have used the main computer. Lumping the computation and terminal costs into one charge makes it more difficult for the engineering supervisor (user) to make a cost effectiveness evaluation and often leads to a distorted picture of the relative cost to do a job.

There are many possible computer and terminal configurations. This study has been restricted to aerospace installations, where, typically, there are large complex problems to be solved that require a large mainframe computer. "Typical" computer computation cost varies between \$250 to \$1,200 per "computational hour." This wide spread results from differences in cost distribution algorithm, computer support personnel in the computer allocation, people or hardware on overhead, and even project accounting or fund distribution methods. The same hardware may cost users different amounts depending on whether it is purchased or leased and whether the lease is long-term, second-party, with or without maintenance, etc. When variations in charges for priority handling (two-hour turnaround, overnight, etc.) are added to this, it becomes apparent that operational cost evaluations must be made on specific circumstances and data.

Although software development costs are not part of the operational cost, the interactive character of an IPAD environment will imply an increased use of interactive programs and will indirectly have an impact on operational costs through increased equipment requirements and software maintenance. Interactive environment programming requires more consideration of the user and his personal involvement during the program design and development than for batch programming. Graphics programs must be smaller to minimize central memory requirements in the interactive mode and will have more actual coding due to the display requirements.

The impact of time and human resource considerations on operational costs within an interactive environment are mainly associated with the time saved in the design process due to shorter problem-solving cycles and reduced turnaround times. The benefits to the user are varied and depend on which of the possible advantages he wishes to exploit. The reduction in time to do a given task may be regarded as an opportunity to exercise some management decision options not available before. Each option has its own payoff and can be selected in light of the long or short term benefits to the project and the company. As a consequence of these options and the present state of development and implementation, a quantifiable impact of the use of interactive terminals on the total design process has so far eluded a firm, empirical definition. Convair's experience has shown that engineering manhours are reduced with the use of interactive environment. Tasks have ranged from one half to one twentieth of the original time. Figure 3-1 summarizes some of the options available to take advantage of the time saved. The assumption made of an average of 5:1 time reduction in production development cycles stems from the Department of Defense Conference on CAD/CAM in Davenport, Iowa (Reference 5); current results confirm this prediction.

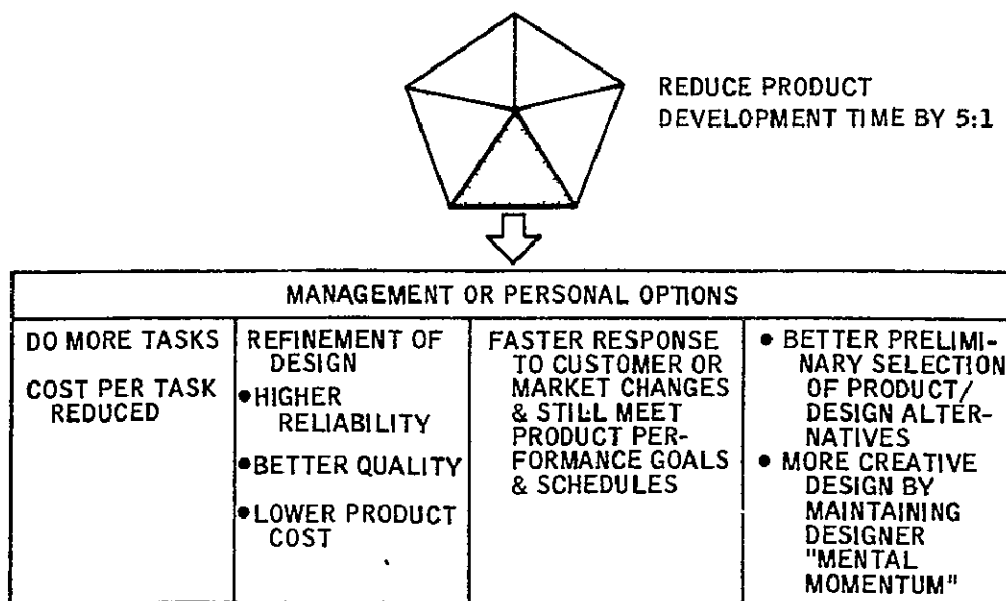


Figure 3-1. Options Available to Use Saved Time

In light of the preceding discussions, it becomes apparent that it is difficult to quantify and estimate projected operational costs within an interactive system such as

IPAD. The only meaningful figures would be "efficiency factors," which could be used to multiply prevailing project costs to estimate projected costs for conducting projects under IPAD. It is clear that a single efficiency factor — one that could be used to multiply total project costs, regardless of the technical complexity of the project — is not possible. Further, it is difficult to quantify cost improvements associated with operation within an IPAD environment, since available benchmarks are inadequate. Assessment of operating costs was undertaken with this realization, as detailed in the following sections.

3.1 Identifiable Cost Increases

There are three areas of cost increase identified to IPAD operations:

1. Computer costs of each Operational Module (OM) execution will increase due to the overhead associated with:
 - a. IPAD's EXECutive
 - b. Data Base Management System (DBMS)
2. Computer costs of each OM execution will increase due to the additional hardware required to support IPAD and due to the corresponding operating system maintenance costs.
3. Interactive terminal costs will add to manhour costs whenever the user is at a terminal.

Each of these factors will be separately treated in the following subsections.

3.1.1 EXEC overhead. - IPAD's EXEC, if used properly, is quite similar to the operating system subutility, which decodes the Operating Systems Control Language (code module 1AJ, Advance Jobcard in CDC's parlance). Operating overhead is probably negligible (e.g., a small percentage).

3.1.2 DBMS overhead. - DBMS overhead is probably significant, particularly in an IPAD environment, which exploits DBMS. Increased costs can be identified with two factors:

1. Substantially increased input/output (I/O) activity.
2. Transfer and manipulation overhead associated with the I/O.

Since the I/O (e.g., disk) activity will probably dominate costs, it was the sole factor considered.

An event analysis was performed (using CDC's design of DBMS) to determine the likely increase in I/O activity using DBMS. Figure 3-2 presents the results of this analysis and illustrates that the most sophisticated DBMS task can increase the I/O activity sixfold over the standard activity (e.g., with conventional FORTRAN I/O).

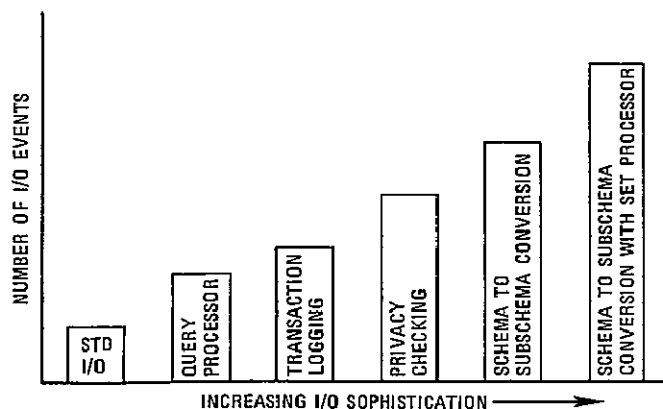


Figure 3-2. I/O Event Analysis Utilizing CDC's DBMS Design

While converting I/O activities to cost, it was observed that these are termed "monitor requests" in CDC's parlance, and Convair uses monitor request in their cost algorithm. (Indeed, this cost algorithm is the result of many years of continuing analysis of the costs affecting computer operations and probably reflects the true marginal cost for I/O activity.) The algorithm is the sum of nine items, one of which is

$(\text{prevailing rate}/3600) (\text{monitor requests}/26.6)$

which figures as a one-percent rate increase per thousand monitor requests.

Figure 3-3 is a histogram of monitor requests for the jobs contained in the OM Questionnaire. (Reference Section 4-3 of Volume IV.) These monitor requests averaged

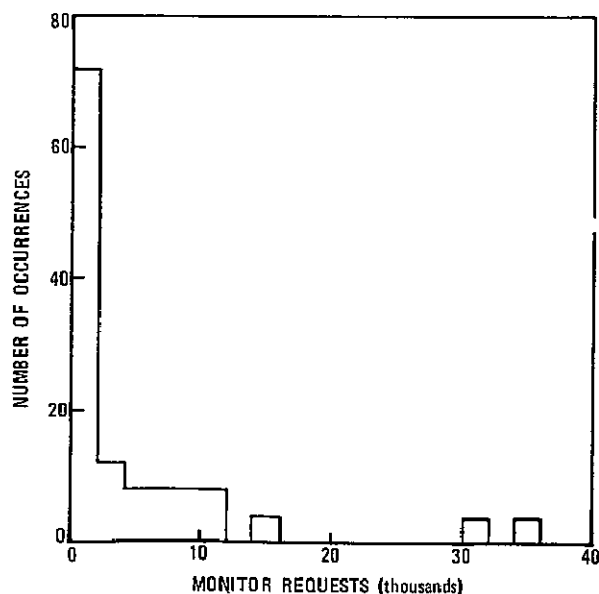


Figure 3-3. Monitor Request per Case

2,100 per case, or an overhead of approximately 2 percent. Some extreme values were noted that could result in a 35-percent increase in overhead; these were jobs with extensive tape copy, however, and are not applicable to the proposed DBMS. A realistic extreme case would be an increase in overhead of about 10 percent.

The overhead probably attributable to DBMS can now be computed. Figure 3-2 suggests that the overhead can vary from two to six times the standard overhead associated with monitor requests. Figure 3-3 (and Convair's cost algorithm) suggests that the overhead per case (i.e., per typical OM) averages about 2 percent

but can realistically range as high as 10 percent. Thus, the average DBMS overhead is expected to range between 4 and 12 percent, but the extreme can range as high as 20 to 60 percent.

3.1.3 Costs associated with increased host computer hardware. - Host computer hardware requirements were identified in Volume IV, Section 5.4. Most major aerospace companies have one of the candidate-class computers dedicated to scientific applications. It is appropriate in this case to estimate the percentage increase in lease costs resulting from the additional hardware required.

Using Convair's current CDC CYBER 70 Model 72 as a base, the following additional equipment must be added to this system to meet the needs of a single project.

1. An additional 32,768 word central memory expansion (Option 10264-3), which is an additional 12 percent* in lease costs.
2. An additional 15M word dual-drive disk storage unit (Model 844-2), which is an additional 1 percent in lease costs.

Thus, it can be assumed that an OM within an IPAD environment could be expected to experience a 13-percent cost increase, other things being equal.

3.1.4 Terminal costs. - The purchase price of Type 4 and 5 terminals was given in Volume IV, Section 5.2. However, many firms lease terminal equipment and charge a "connect time" rate based on a 60 percent utilization factor for the prime shift. Thus, the use of a given type of terminal costs the user a fixed rate per connect hour.

The connect time rate can be compared with prevailing direct labor rates to assess how much more efficient a user must be to amortize his terminal costs while connected. Such a calculation using a lumped average labor rate suggests that a 40-percent improvement in efficiency is required with a Type 4 terminal and an 80-percent improvement with a Type 5 terminal. Experience of individuals who have worked with these devices indicates that these improvements are easily achieved.

3.1.5 Conclusions. - Estimates of cost increases have identified the required expansion of the host computer hardware with a 13-percent overhead and DBMS with an overhead that averages between 4 and 12 percent, but could climb as high as 20 to 60 percent at an extreme. It is obvious (upon reflection) that not every FORTRAN file (e.g., SCRATCH files) should be handled by DBMS. Rather than a quantitative cost increase, what has been identified here is the caution that must be employed with the insertion of existing OMs.

*These are figured on a comparable basis of "book" lease costs, including maintenance but excluding taxes and applicable discounts. Only the core complex is included. Remote entry stations are lumped with terminals.

Since almost every file is associated with the smaller OM's (there are usually only two files) on DBMS candidates, it is unlikely that the 60-percent figure could ever be reached. IPAD operation could average a 25 percent increase (13 percent + 12 percent) per OM execution, however.

3.2 Identifiable Cost Decreases

Most costs associated with human resources should decrease in an IPAD environment:

1. The manhour costs associated with each OM execution will decrease due to:
 - a. Use of the EXEC and task control sequences (TCS) for tasking.
 - b. Use of the General Purpose Utilities for:
 - Configuring the User File data subbase (Query Processor)
 - Collecting and storing the task initiation data (Query Processor)
 - Examining OM results (Query Processor and General Graphics Plotter)
 - Optimizing or parametrizing (OPTUM)
 - Graphing (General Graphics Plotter)

These costs are partially offset by the computer costs associated with executing the General Purpose Utilities.

2. The manhour costs associated with each design task will decrease due to:
 - a. Use of the SCHEMA assembler for task integration.
 - b. Use of the EXEC's RECORDER for saving TCSs for later re-execution (e.g., with an updated data base).
 - c. Use of the EXEC's WRITER for constructing general purpose TCS skeletons from specific TCSs.
 - d. Use of the EXEC's EXPANDER to construct task-specific TCSs from general TCS skeletons.
 - e. Use of TCSs to construct and review presentations of design data (which is temporarily stored in the data base).
 - f. Use of project review files to communicate analysis (e.g., trade study) results in a simple yet highly effective manner (similar to a viewgraph presentation and/or a movie).
3. Manhour costs associated with project management will decrease due to:

- a. Close management control of all official project data.
 - b. Increased visibility afforded by:
 - Instant availability of all current design data.
 - Timely communication to all levels of the project structure via task status/action files.
 - Effective communication on current evaluation of the design through the project review files.
 - Availability of an indepth track of any user task via its User Task Trajectory.
 - c. Decreased vulnerability afforded by:
 - Effective control of data access (privacy).
 - Roll-back of the data base (recovery).
4. The man-hour costs associated with OM development will decrease due to the availability of existing FORTRAN OMs to the IPAD (once inserted via the DML Insertion Preprocessor and the SUBSCHEMA Assembler). This savings will be negligibly offset by the cost of executing these special-purpose utilities.
 5. Although not usually associated with cost, the reduction in design time discussed in Volume IV, Section 5.1 should result in additional cost savings.

3.3 Summary

Substantial cost savings can result if:

1. The user (engineer) is adequately loaded with parallel or quick-reaction tasks to offset his propensity to over-engineer.
2. Rising computer costs do not negate the expected man-hour savings.

The first is a problem that only the project management can solve. The second can be mitigated by the creation of efficient — in both time and core requirements — code associated with DBMS. This responsibility rests with the host computer manufacturers.

4 IPAD SYSTEM BENEFIT ASSESSMENT (PHASE II, TASK 6)

In Volume II, Phase I, Task 1, Section 3 a typical aerospace vehicle design process was examined by having a competent engineer from each of the selected technical disciplines explain how he performed his task and the tools he used to do it. The design process was seen to consist of two basic tasks, namely

1. Creative tasks (e.g., management planning, vehicle configuration design).
2. Data generation and manipulation tasks (i.e., routine analyses, trade off studies, and sensitivity studies).

Although this design process "works," it contains some deficiencies; correction of or mitigating these deficiencies by an IPAD-type system would result in tangible and intangible benefits as described hereunder in this section.

The major deficiencies in the design process noted above, and it is believed that these deficiencies exist to a greater or lesser extent in all aerospace companies, are as follows:

1. It is becoming increasingly difficult to meet Customer requirements for designing higher performance vehicles in a shorter period of time and at a lower cost.
2. Irrespective of whether one considers the Conceptual Design, Preliminary Design, or Detailed Design phase, quite frequently some significant disciplines do not get their inputs into the design process with the required emphasis in a timely manner.
3. Notwithstanding efforts to the contrary, some tasks are still discussed and treated as if no "common thread" existed between the various disciplines (i.e., the engineer in each discipline tends to think that he is the one who is steering the log that is floating downstream); the consequence is that computer programs are operated as separate entities in individual disciplines.
4. In some cases, inconsistencies exist in different disciplines both in outputs and inputs to computer programs using the same type of data, and also there is some duplication of subfunctions among computer programs.
5. There is lack of timely visibility in some of the evaluation processes because man is not interactively in the loop during key portions of the problem-solving procedures.

The impact of these difficiencies on the aerospace vehicle itself is a combination of one or more of the following effects:

1. Degraded performance (i. e. , primarily because all significant sizing, evaluation, and optimization studies are not performed in a timely manner).
2. Increased costs (i. e. , primarily because the studies are not performed in a cost-effective, integrated manner and significant amounts of drudgery remains within many creative and evaluation procedures).
3. Inability to meet some of the short time schedules imposed by the Customer (i. e. , primarily because the response times for disciplinary group evaluations and properly interfaced team efforts are not fast enough and commensurate with the magnitude of these new tasks under present management/engineering team modus operandi).

4.1 Impact of IPAD on Performance of Aerospace Vehicles

The impact of an IPAD-type system on the performance of aerospace vehicles can best be illustrated by describing recent experiences (i. e. , within the last year) at Convair Aerospace Division. These experiences relate to:

1. A STOL tactical aircraft study
2. A Navy V/STOL fighter attack aircraft
3. A Space Shuttle launch study
4. A Large Space Telescope study

4.1.1 STOL tactical aircraft study. - A STOL tactical aircraft study was performed for the USAF under Contract F33615-71-C-1754; results of this study are shown in Reference 6.

This study, which is typical of a military or commercial transport, was concerned, in part, with defining a configuration (i. e. , the external geometry) that would perform a specified STOL tactical mission in an optimum manner with minimum takeoff gross weight. Numerous parametric studies were made involving airplane geometry and engines. Figures 4-1 to 4-3 illustrate end results of typical tradeoff studies performed to assess the impact of configuration and performance parameters on the design takeoff gross weight of this STOL aircraft.

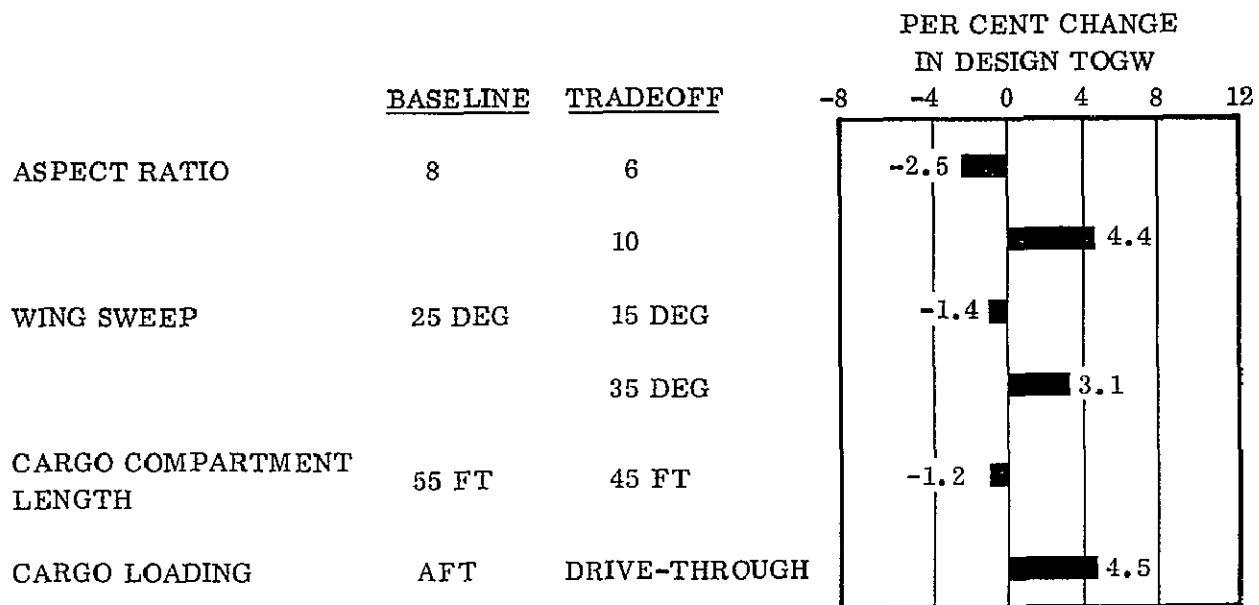


Figure 4-1. Configuration Design Tradeoffs

The results of these tradeoff studies and others were used to arrive at the final vehicle design. Table 4-1 shows a comparison of parameters for an early design and the final design. These examples clearly show that an adequate number of matrix or parametric studies must be made to achieve optimum performance with minimum weight.

TABLE 4-1. STOL TACTICAL AIRCRAFT EARLY & FINAL DESIGN COMPARISONS

PARAMETER	EARLY DESIGN	FINAL DESIGN
Engine	GE13/F2B	GE13/F2B
Wing Area (ft ²)	1,550	1,550
TOGW (lb)	148,200	137,450
Mid-mission Weight (lb)	134,200	125,700
Rated Thrust (lb)	18,600	15,075
T/W	0.555	0.480
W/S (lb/ft ²)	86.6	81.1
Takeoff Distance (ft)	2,000	2,000
Landing Distance (ft)	990	1,530

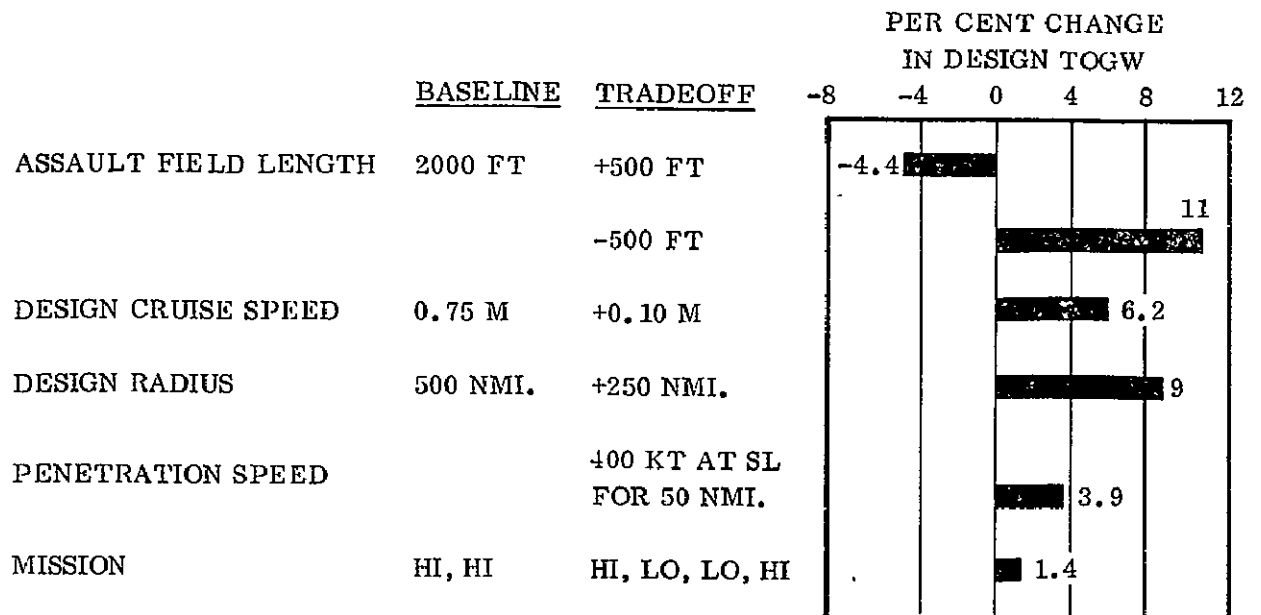


Figure 4-2. Performance Design Tradeoffs

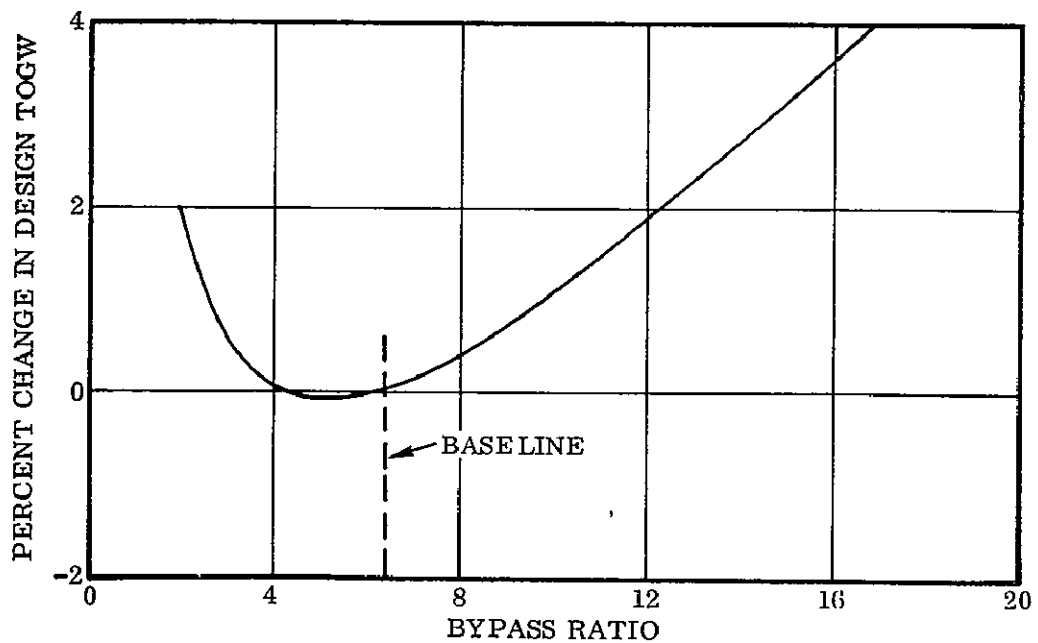


Figure 4-3. EBF Bypass Ratio Tradeoff

The procedures for making these studies are straightforward and can be readily accomplished on a digital computer using either a batch-mode operation or an interactive graphics mode operation. Table 4-2 shows a comparison of typical batch-mode and interactive-mode operations to obtain the data from which Figure 4-1 was plotted. Although this comparison is for one matrix computation it is clearly evident, by any standard of measurement anyone cares to choose, that the interactive graphics mode proposed for IPAD saves time and money; an aerodynamics engineer and a mass-properties engineer worked closely together in the interactive graphics mode to attain enhanced problem-solving visibility and to achieve the noted savings. When the interactive mode is extended to other single and multi-disciplinary investigations the time and cost savings are very tangible and substantial. An intangible benefit resulting from the savings in time is that engineers have more time to perform creative tasks.

TABLE 4-2. COMPARISON OF BATCH MODE AND INTERACTIVE MODE OPERATIONS. STOL TACTICAL AIRCRAFT

Item	Batch Mode	Interactive Mode
Number of Computer Entries and Exits	10	1
Man Hours Required to Perform One Matrix Study	12	1

4.1.2 Navy V/STOL fighter attack aircraft. - Over a calendar time span of 50 days, classified work was performed on a Navy V/STOL fighter attack aircraft.

This study illustrates a case where the customer specified a set of requirements and ground rules which necessitated developing new external geometry and numerous technical analysis in a very short calendar-time span. A specific customer request was to obtain the steady-state and transient loads due to a salvo store ejection. Considering the available calendar time span and the time required to establish the aircraft external geometry, mass properties, and stiffness properties this request would normally have been executed by computing steady-state loads and applying an arbitrary factor (i. e., based on experience) to these loads to account for the transient loads. This, of course, is a very crude approximation to the actual transient loads. However, interactive graphics equipment and programs were used to quickly assemble and debug a NASTRAN model of the aircraft. This model was used to obtain the aircraft vibration modes and frequencies which subsequently were used to obtain the actual transient loads

due to salvo store ejection as reported in Reference 7. It is noted that the original finite element model had a number of errors in it (e.g., grid or mass points were improperly located, and improperly connected by springs.) These errors were readily detected and corrected in approximately thirty minutes using the interactive graphics capability; the analysis of the corrected model produced correct vibration modes and frequencies the first time it was run on the computer. If batch mode operations had been used, it would have taken at least three days elapsed time to have corrected the model and obtained correct vibration modes and frequencies, and time and money would have also been wasted on the computer obtaining incorrect runs. Similar models, but for different weight configurations, were also debugged for computation of vibration modes and frequencies that were used in the computation of actual transient landing loads and flutter speeds.

Thus, notwithstanding a very tight schedule imposed by the customer, the above example shows that use of an IPAD-type capability permitted computation of realistic strength and stiffness requirements which resulted in creditable weights, and also saved time and money. Such tangible benefits are multiplied when IPAD-type systems are applied to solution of other dynamics problems, and to problems in other disciplines. Intangible benefits from the time savings is that it gives the engineers more time to perform creative tasks, and it frees the computer to do other work instead of using it to "laboriously debug" mathematical models (i.e., by executing possibly several expensive runs with erroneous input.)

4.1.3 Space shuttle launch study. - During the Space Shuttle Phase B study under North American Rockwell Contract MON 7 BMX-587600H Convair used launch computer program P5458, which computes ascent trajectories in the presence of winds aloft, for vehicle load and control studies. The use of this program involved debugging input data, baseline synthesis, parametric studies, and presentation of results.

During the course of this study many computer runs had to be made as the vehicle and control configuration evolved. Figure 4-4 shows typical results in the form of an engine deflection versus time history. Both the batch mode and interactive mode were used in these studies. Table 4-3 shows a comparison of batch-mode and interactive-mode operations to obtain the results shown in Figure 4-4. For the total launch study runs made over a period of one year it was estimated that the batch mode would have cost \$101,400 and the interactive-graphic mode cost was \$7,410. Thus, the tangible cost saving is self-evident. The intangible benefits are that similar savings can be achieved on other tasks when interactive graphics is used.

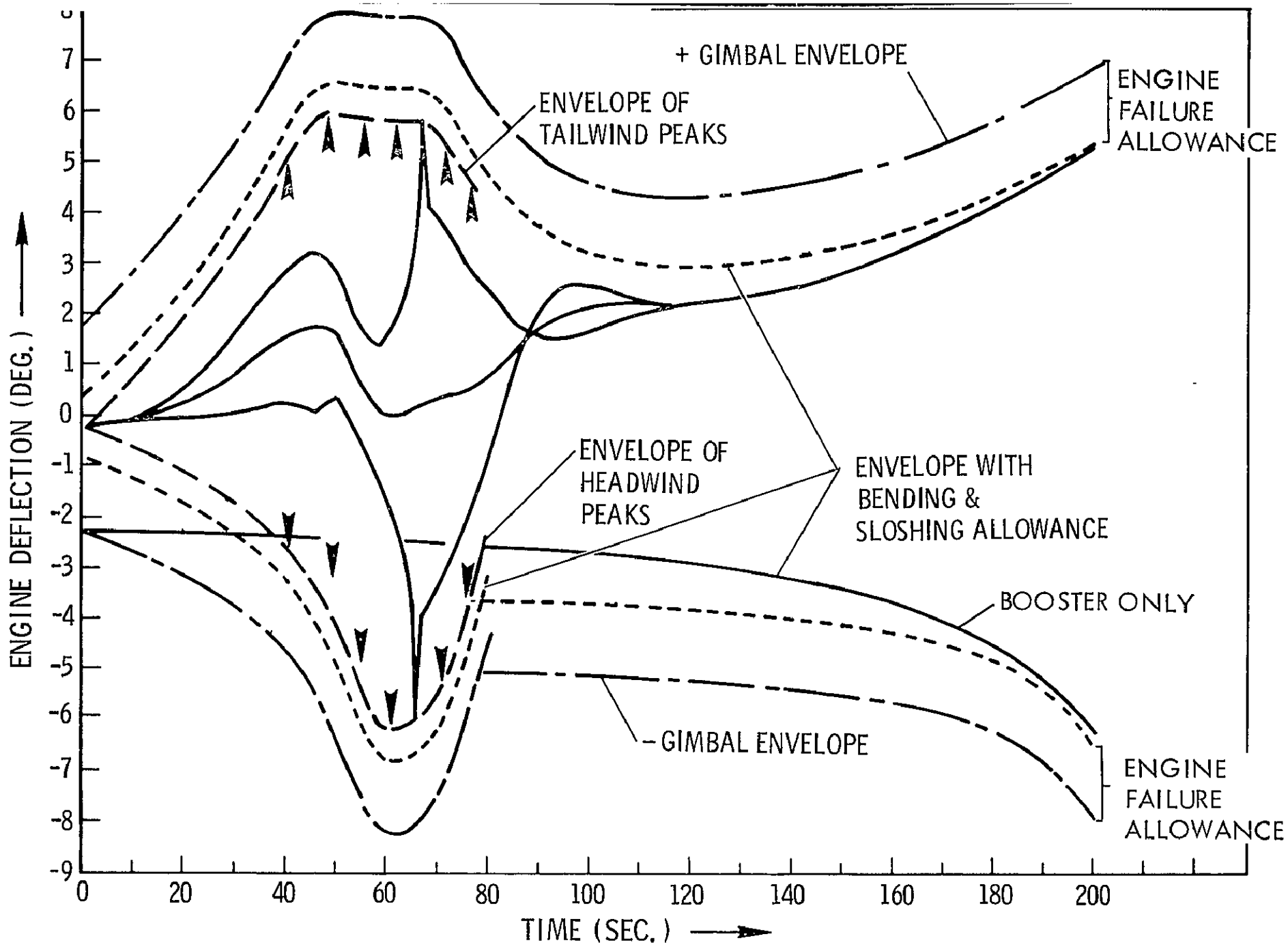


Figure 4-4. Space Shuttle Engine Deflection-Time History

TABLE 4-3. COMPARISON OF BATCH MODE AND INTERACTIVE
MODE OPERATIONS SPACE SHUTTLE LAUNCH

Item	Batch Mode	Interactive Mode
Number of Computer Entries and Exits	6	1
Elapsed Hours to Perform One Study	100	2

4.1.4 Large space telescope study. - Under NASA Contract NAS8-27539 a study was made of thermal distortions of a large space telescope. Satisfactory alignment of the optical elements required that these distortions be accurately known and that they be small. Results of these investigations are shown in Reference 8.

The NASTRAN mathematical model used was a complex array of panels and trusses. An interactive graphics program was used to debug the model (i.e., verify the mass locations in three dimensions, and verify the connectivity of the structural elements). "Exploded views" of closely clustered panels were projected on the graphics console to provide good visibility for the debugging process. Table 4-4 shows a comparison of batch-mode and interactive-mode operations to perform a typical temperature analysis. As previously shown the tangible benefits of reduction in time and cost using IPAD-type interactive capability are readily apparent. An intangible benefit is that the prompt debugging of complex mathematical models allows the engineer to use the savings in time for other creative work.

TABLE 4-4. COMPARISON OF BATCH MODE AND INTERACTIVE
MODE OPERATIONS. LARGE SPACE TELESCOPE.

Item	Batch Mode	Interactive Mode
Number of Computer Entries and Exits	3	1
Elapsed Hours to Perform Typical Temperature Analysis	50	6

4.1.5 Overview. - In summary , an IPAD system is expected to have the following impact on studies, which in turn will result in improved vehicle performance:

1. IPAD with man in the loop will allow faster analysis, will permit more comprehensive vehicle and subsystem sizing and optimization, and will permit more complete tradeoff studies with an integrated participation of various disciplines.
2. Higher management confidence that the resulting design will be better because all groups will work from a common data base thereby eliminating or reducing inconsistencies, because IPAD's faster capability makes it easier and faster to evaluate alternate design solutions, and because the selected design will have better reasoning and backup behind it.
3. The effects of design changes on configuration, performance, and costs will be easier to predict.
4. Although opinions vary on how much design calendar-time can be saved and how much design costs can be reduced by using IPAD, the calendar time for design freeze has to be reduced and the costs decreased because of the quicker iteration loops possible with IPAD.
5. Should enable zeroing in on the best design more quickly allowing more detailed study of the selected design (which becomes a larger and more pressing requirement as speed, altitude, maneuver, complexity, size, power, etc. increases).

4.2 Impact of IPAD on Engineering Work

Convair has had both functional and project organizations for many years. In both types of organizations, with allowance for overlap, the aerospace engineering work was primarily accomplished as follows:

1. During the 1940's the engineer in each technical discipline sat at his desk and performed analyses in his field with the aid of a slide rule and/or small desk calculator. The engineer in each design discipline also sat at his desk and made layout or detail drawings in his field with the aid of parallel bars, triangles, rulers, and drafting boards. Most of the vehicles were subsonic and some, in the late 1940's, were supersonic. Cook-book formulas and conservative design approaches were frequently used, and parametric studies were minimal, because of the time required to make analyses. Optimization and sensitivity studies were practically unheard of. Communication between disciplines was mostly verbal. Each engineer had good visibility pertaining to his task, but he did not always have this visibility from a systems point of view.

2. In the 1950's the engineer in each technical discipline began to make much greater use of large-scale digital and analog computers, and the engineer found himself in the computing facility just as often as at his desk. Design engineers began to use drafting machines to make layout and detail drawings more rapidly. The vehicles were subsonic, supersonic, and in the late 1950's extended to the hypersonic speed range. It became mandatory to perform more parametric studies, and more optimization and sensitivity studies to design satisfactory high performance vehicles. Communication between disciplines started to become more formal (i. e., written). However, in the computer batch-mode operation the engineer frequently lost visibility; this usually resulted in computer reruns to correct errors or to achieve satisfactory optimization. Furthermore, a turn-around time of one week to perform a set of computations was not unusual. Also, since each discipline was generally preoccupied with its own operational modules, the engineer did not have very good visibility from an integrated system point of view.
3. During the 1960's the use of improved large-scale digital computers became a way of life with the engineer in each technical discipline finding himself in the computing laboratory quite frequently. The engineer was concerned with reducing the turn-around time to one-day or less, and this had been achieved. He was also concerned with obtaining better visibility that his mathematical model was correct before making a computer run; this he achieved by using interactive graphics. Design engineers, however, continued to use drafting machines, and did not make extensive use of interactive graphics. The vehicles covered the same speed range as those in the 1960's but higher performance was demanded of them which in turn demanded more analytical investigations. Communication between disciplines was formal, and design reviews were formally conducted. Although the engineer attained much better visibility in his discipline, he was still primarily preoccupied with his own operational modules and, typically, did not have great latitude and visibility from an integrated systems point of view.
4. In the 1970's Convair and other aerospace companies are striving to improve their engineering work by eliminating or mitigating the primary deficiencies listed in the beginning of Section 4. These improvements pertain, in part, to economical IPAD-type interactive graphics systems as noted in Reference 9 . However, considerable effort and funds are still required to implement an IPAD system as defined in this study.

IPAD is ideal for tasks requiring the generation, storage, and manipulation of large volumes of data. Typical examples are engineering team evaluation and design tasks associated with an aircraft project, such as vehicle configuration and subsystems

design, repetitive routine analysis, tradeoff and sensitivity studies, etc. The need to access large data bases and operate with a multitude of data subsets is typical of project tasks in all disciplines. This requires streamlined data accessibility and communication capability such as provided by an IPAD system. Furthermore, an improved problem-solving engineering capability will be fostered by IPAD, resulting in shorter evaluation schedules and reduced engineering costs. Thus, IPAD is seen as having the following near-term and far-term impact on engineering work:

1. Designers and engineers will have a more prominent participation in the design process by efficiently operating and controlling the IPAD system. Creative and evaluation functions will be performed much more economically with IPAD than with a batch-mode operation.
2. The quality of tradeoff and optimization studies will improve because the analyses will be integrated and cover more significant interdisciplinary variables. Studies of "growth versions" of vehicles at any time (i.e., during the design of the initial vehicle or after it has been in service for some time) can be performed very economically within an IPAD system.
3. A "Data Bank Administrator" will be required within a project operating with an IPAD system to force consistency of the data base throughout all engineering working groups, and to ensure proper identification and storage of project data.
4. For analytical groups (e.g., aerodynamics, dynamics, and mass properties) it will force the output of the operational module of one discipline to be consistent with the input required by the operational modules of other disciplines, which will save time and money through improved communications.
5. It will cause the routine drafting tasks to become more automated while maintaining the creative design tasks under the control and proficiency of the user.
6. For the test groups the traditional formats of data reduction and presentation could be automated so that the data output can be readily entered and evaluated within an IPAD system. A desirable goal is to reduce to twenty-four hours or less the time span from recorded test data to useable reduced test data by the cognizant discipline; diligent IPAD effort should be able to accomplish this. To illustrate the significance of this, at the present time the reduction of measured wind tunnel pressure data into a useable form can take up to two weeks.
7. Estimating tasks will tend to use grass-root approaches and will become more automated.

8. Special management operating modules and automated status reports will give management greater visibility, and will enable management to make decisions in a timely manner, consistent with shorter design schedules envisioned for IPAD.
9. The skill mixture in the engineering department will change (e.g., the number of engineers and engineering aides should decrease, but the number of engineers who can perform their work within an IPAD environment would increase; the net effect would be a reduction in total manpower).
10. Report preparation can be simplified by the hard-copying feature of interactive graphics and the need for reports can be reduced by storing the data in the project data banks, available to all users.
11. Since IPAD will effectively cope with the data generation and manipulation tasks, implementation of IPAD will give the engineer more time to devote to creative tasks.

4.3 Disciplinary Interfaces

The introduction to Section 4 listed five major deficiencies in the current design process. At least three of these are strongly due to inadequate consideration or lack of timely data communication among interfacing disciplines. An IPAD system will substantially mitigate these problems because it will:

1. Require consistency of the data base in all disciplines through the "Data Base Administrator."
2. Require the clear definition of all interfaces because IPAD operates as an integrated system.
3. Require sizing and optimization studies to be considered from a wider interdisciplinary point of view rather than from each single discipline viewpoint. Personnel in each discipline will have to become fully aware of how their inputs and outputs affect the total system.
4. Require analytical, design, test, and management personnel to work in a timely manner as an integrated team.

It is evident that the tasks noted above do get done by the current design processes; however, they get done in a fragmented manner. IPAD will accomplish these same tasks in an integrated manner with improved efficiency and economy.

In the IPAD system the currently accepted concepts of interface problems (e.g., the engine location from the standpoints of drag, noise, and flutter) will broaden in

scope to include new interface items such as the rapid reduction of test data into a useable form.

Solution of interface problems in IPAD will also result in an upgrading of personnel by increasing their interdisciplinary capability. This greatly benefits manpower planning and assignments in companies or organizations that execute multiple programs simultaneously.

4.4 Answers to NASA's RFP Questions

This section presents answers to questions posed by the NASA in relation to Task 6 and to questions added by the contractor. The following subsections present the questions followed by the answer to each of them.

4.4.1 What tangible evidence do you have that would suggest an IPAD system would improve performance of military aircraft or return on investment of commercial aircraft? - Figures 4-1 to 4-3 and Tables 4-1 and 4-2 in this volume, which pertain to a military aircraft but which could be equally applicable to a commercial aircraft, illustrate how an IPAD type operation was used to define an aircraft configuration to perform a specified STOL tactical mission with minimum takeoff gross weight (TOGW). The final design TOGW was 137,450 lb. versus the early design TOGW of 148,200 lb; here both configurations would perform the mission, but obviously the performance and return on investment of the final design are noticeably better than for the early design. This example also implies that an adequate number of parametric studies must be made to achieve optimum performance with minimum weight, and that the reduction of the number of parametric studies, because of time or cost limitations, can easily result in higher weights than necessary to achieve the same performance requirements.

4.4.2 Given the present engineering work organization, what is the likelihood that engineers will be able to do more creative work when tedious and routine tasks are taken over by IPAD? - Tables 4-2 to 4-4 illustrate typical times saved by using IPAD-type features (e.g., the interactive mode) on three specific studies. Engineers who worked on these studies were asked what they did with the time saved. Their replies were as follows:

1. It enabled them to vary significant parameters over a wider range and thus do a better technical job.
2. It enabled them to do more discrete tasks.
3. It enabled them to perform several tasks in parallel rather than in series.
4. It gave them more time to think and do creative work (e.g., consider alternative design solutions and operational concepts to reduce development and service operation costs).

4.4.3 Will the system bring closer cooperation between the people from different disciplines? With what results? - Detailed answers to these questions can be found in Section 4.3 of this volume.

4.4.4 (Added by Contractor) How can IPAD do creative work? - IPAD itself does not do and is not intended to do creative work; IPAD will enhance the performance of design and data generation and manipulation tasks permitting the user to respond much faster and more economically than done heretofore. However, creative work, creative ideas, and esthetic tastes are the drivers behind the user exploiting the IPAD system capability.

4.4.5 (Added by Contractor) The aerospace industry got the job done before without IPAD, so why is it needed now? - From the striking of flint to the harnessing of atomic energy, man has explored and utilized many latent forces of nature. He has continually evolved new technology to better exploit those forces and the power of his creativity. In the past few years, man's creativity has imparted momentum to a new technology. Many of its features are already here, have proven themselves cost-effective, and have been accepted as the new ways of performing many design tasks. IPAD will serve as the vehicle for coordinating and integrating the development of such technology, it will prevent duplication of effort and government funding, and it will contribute to the conservation of national resources.

5 IPAD SYSTEM IMPACT ON COMPANY ORGANIZATION (PHASE II, TASK 7)

Although no two aerospace companies have the same organizations, they all fall within one of two basic types, namely a functional organization as illustrated in Figure 5-1, or a project organization as illustrated in Figure 5-2. As the names imply, a functional organization services all projects whereas a project organization usually services only a single project. Each type of organization works. Each has its advantages and disadvantages. In many companies (e.g., Convair Aerospace Division) both types of organizations are actually used at the same time in an attempt to maximize the advantages and minimize the disadvantages.

Once the basic type of organization is selected, the details of the organization used are usually defined on the basis of people and their capabilities rather than on specific tasks. This is why design loads are obtained in the Dynamics group in one company, in the Structures group in another company, and in the Aerodynamics group in still another company.

In line with its flexibility, the intention is to implement IPAD within the existing functional or project organizations in any company, with the proviso, however, that such existing functional or project organizations "bend a little" in order to exploit IPAD features. What is involved in this "bending" is described in the following sections.

5.1 Ramifications of Organizations Relative to Acceptance of IPAD

Through experience over the years each aerospace company has accumulated good policies, procedures, and practices that it has compiled in the form of manuals; Figure 5-3 is a typical list of such manuals used at Convair Aerospace Division. Each of these manuals came about because there was a need for it. Not only do these manuals prevent "reinvention of the wheel for each new project," but they have been developed because "necessity is the mother of invention"; top level management policy specifies that these manuals are official and directs that the policies, practices, and procedures therein be implemented in the company programs. If any item in any manual is found not quite applicable when tried with a new program, then the item is modified or expanded. These manuals are revised as required to keep them current.

An examination of elements of the aerospace vehicle design process as discussed in Phase II, Task 6, indicates the need for an IPAD system. Its implementation calls for an IPAD manual, and a top level management (i.e., President, Vice President, and

Division Manager) policy that specifies that the IPAD manual is official and the circumstances and conditions under which IPAD shall be implemented on its programs. Each company would have flexibility in preparing its IPAD manual in the same manner that it now has in preparing any of its manuals (e.g., the design manuals).

Sanction of IPAD by top level management has the following significant implications:

1. It requires that the responsibility of each group (i.e., engineering or business) in IPAD be defined in the company "Standard Management Practices" manuals and that this responsibility be executed.
2. It requires that management/engineering members of a project team have access to "balanced" capabilities mounted on the IPAD system to enable smooth operation and timely interfacing between engineering evaluations and management decisions.
3. It will reduce redundant operational modules and it will streamline the design process.
4. It requires that management and users of IPAD contribute to company plans for "Capital Facilities" requirements for IPAD.
5. It requires implementation of training programs to instruct people on the IPAD system features and to train people to use IPAD effectively. An IPAD educational course should be organized and implemented for these purposes.
6. It allows IPAD to compete for company Independent Research and Development (IRAD) funds to improve it.

It is appropriate to address ourselves to the question "When should IPAD be used, or when should IPAD work begin?" As repeatedly stressed in this study, IPAD is an ideal tool for solving problems where a very large volume of data is generated and manipulated, particularly in a repetitive manner. Many of the IPAD features and capabilities (e.g., data base management system, general purpose utilities, interactive graphics environment) can be used on a stand-alone basis by any single user. The IPAD system, though, was born from integrated-team-effort needs and thus it has been designed such that project team members can mount on it whatever management and engineering operational modules are needed for a project, at a given time. In short, the IPAD features and capabilities will be available irrespective of the user being a single individual or a project team. The larger payoff is expected, though, in team efforts where the project data management and integrated-multidisciplinary-effort needs are beyond a certain threshold or "critical mass." In the early stages of evaluating a new idea and before it evolves into a "project", there may not be a need to mount a team capability on IPAD since the "team" has not yet been formed; individual

FOLDOUT FRAME

ORIGINAL PAGE IS
OF POOR QUALITY

GENERAL DYNAMICS Convair Aerospace Division San Diego Division		NO. C16 REVISION 27 BY BCK APPROVED H. E. Newton VICE PRESIDENT-PROGRAM DIRECTOR	DATE 12 PAGE 1 OF 1 EFFECTIVE 4-15-73
LAUNCH VEHICLE INSTRUCTION SUBJECT LAUNCH VEHICLE PROGRAMS ORGANIZATIONS			

FOLDOUT FRAME

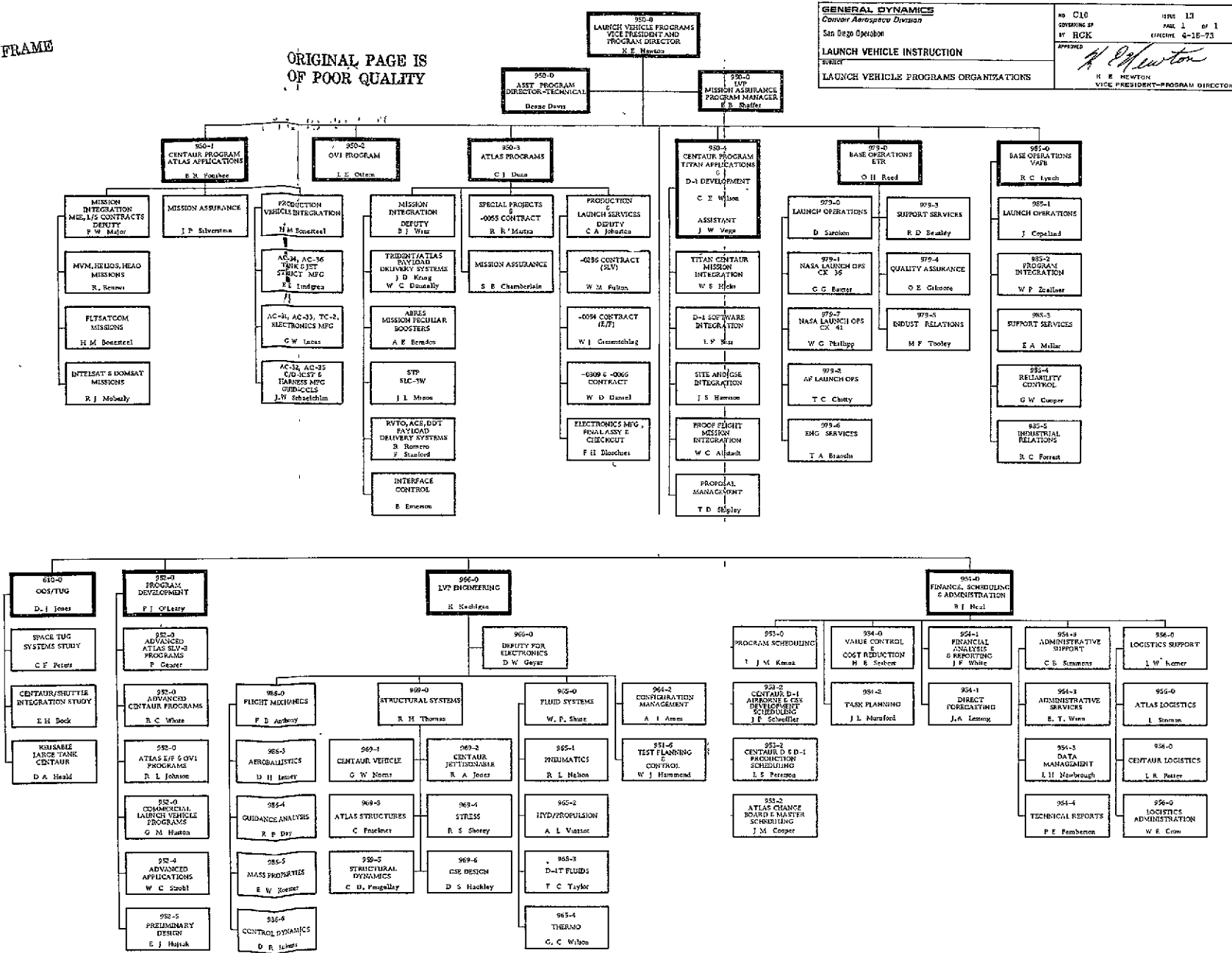
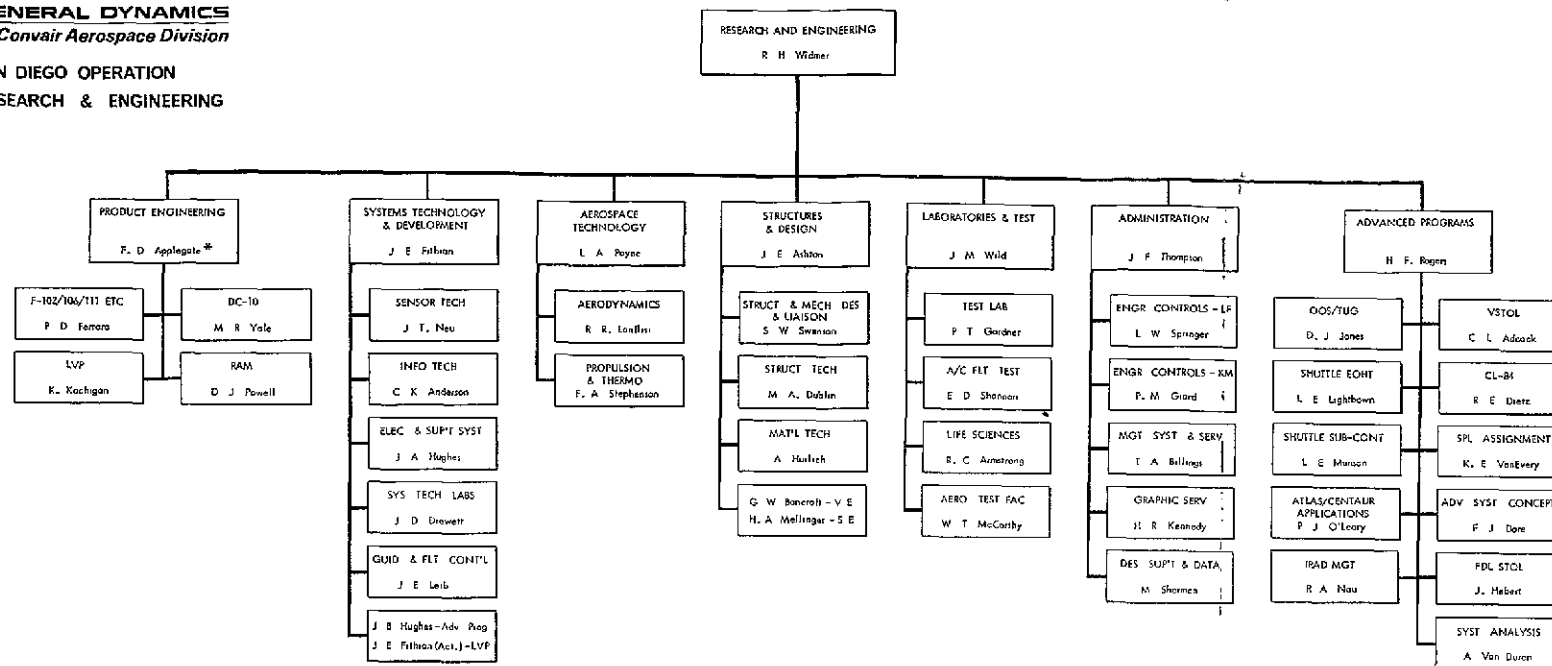


Figure 5-2. Project Organization Example Chart

ORIGINAL PAGE IS
OF POOR QUALITY**GENERAL DYNAMICS**
*Convair Aerospace Division*SAN DIEGO OPERATION
RESEARCH & ENGINEERING

*Deputy in R. H. Widner's Absence

APPROVED *R. H. Widner*
22 May 1972 R. H. Widner

Figure 5-1. Functional Organization Example Chart

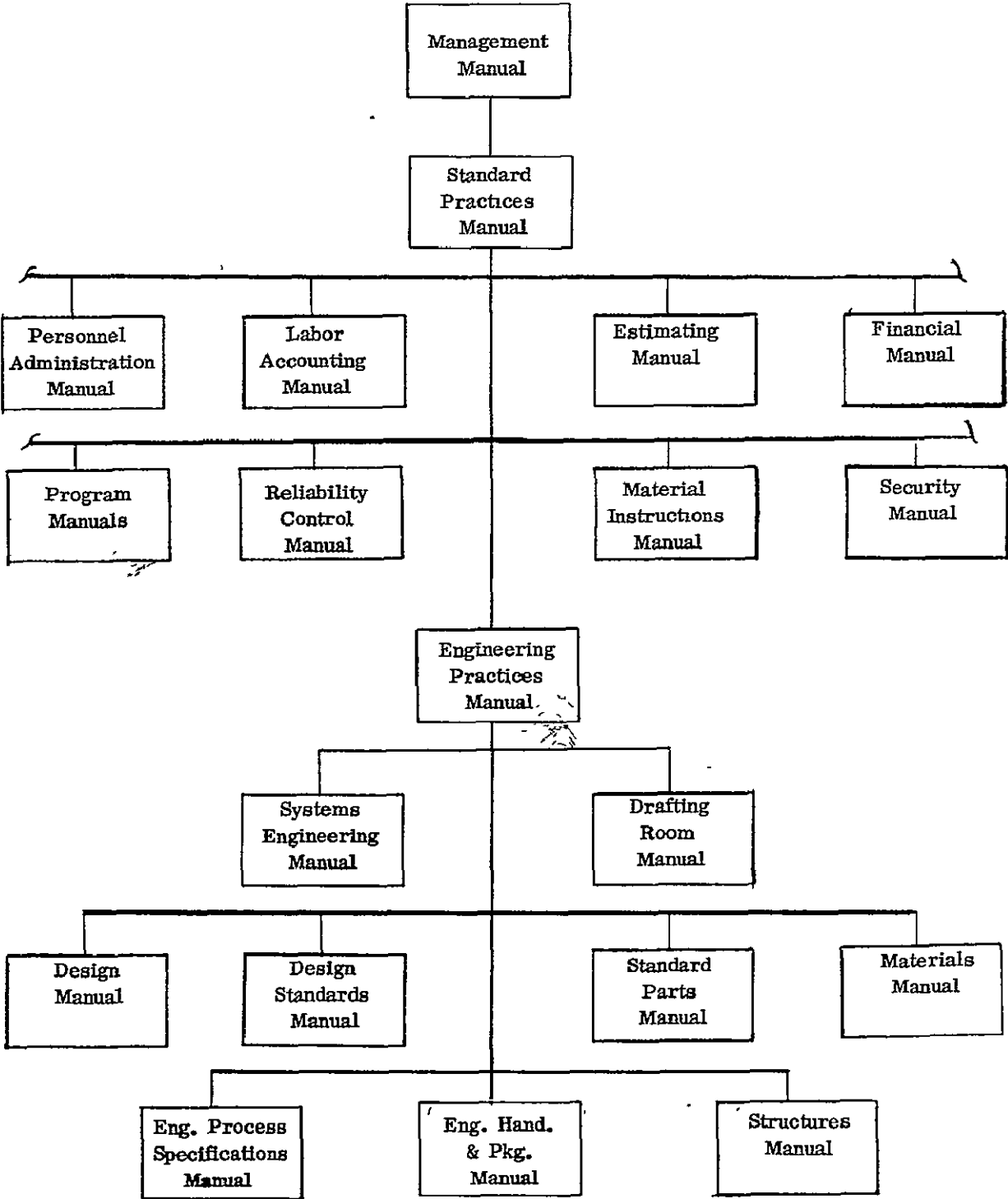


Figure 5-3. Division Manuals Relationships

users, though, can nevertheless operate under the IPAD system with whatever capability they install on it to evaluate the new idea. Virtually every project starts out with tasks involving creative thoughts, customer contacts, specifying ground rules, policy decisions, preliminary marketing surveys, intangibles involving unique experience, rough sketches, back-of-the-envelope type of computations, capital facilities requirements, probable funding, and similar tasks; IPAD may not be needed and should not be involved in many of these tasks; however, outputs from execution of individual tasks can be and should be used in IPAD for effective conceptual, preliminary design, and detail design studies.

Having the aforesaid remarks in mind, implementation of IPAD could be accomplished by existing company functional and project organizations as illustrated hereunder. Inasmuch as IPAD is flexible, each company could implement IPAD with some variations of the illustrations shown herein.

To assist in executing a given project, current functional organizations furnish the required technical and business manpower, facilities (i e , computers, wind tunnels, etc), and test equipment. Functional organizations will continue to do this in an IPAD environment, but additionally they will accomplish the following tasks

1. Develop, adapt, maintain, and improve the IPAD system and its components One example of this is the automatic reduction of raw wind tunnel data in a form that is readily usable by aerodynamics, thermodynamics, and dynamics operational modules.
2. Train and/or assist in training personnel (i e , management, engineering, and business) in the coordinated use of IPAD as a working tool. This training must emphasize that IPAD operation is a team effort to achieve an integrated design; each member of the team must understand how his operational modules interface with the operational modules of other members of the team The training must also teach "when to use IPAD on what projects" and "when not to use IPAD and on what projects"; in other words, man will control IPAD and not vice versa
- 3 Supply a Data Base Administrator and other technical and business personnel to a project office to assist in the implementation of IPAD for the project. Inputs of functional supervisors to a project would nominally be through the IPAD -trained personnel they supply to the project Any conflicts that could not be resolved with this arrangement would be handled by contacts between the functional supervisor and the project office

Actual execution of projects in most companies is usually accomplished by a project organization. The use of IPAD is best suited to a project organization where

it is required to respond quickly in an integrated manner. Figure 5-4 shows a planned Convair Aerospace Division VTOL aircraft project organization (i.e., if the block showing the "Data Base Administrator" were omitted); to implement IPAD in this typical project organization would require only.

1. The assignment to the project of personnel from functional groups who understand IPAD and are dedicated to making it work efficiently for them.
2. The addition of a "Data Base Administrator" who reports directly to the Program Director.
3. The use of automated management techniques to permit timely interfaces with quicker engineering evaluation cycles and shorter response times made possible with IPAD.

Where existing procedures and practices specified in company manuals fit into the IPAD system, they will be used. Where these procedures and practices can be enhanced by the IPAD system, they will be modified as required. Typical examples of where existing procedures and practices could be changed to best exploit the IPAD system and streamline the design process are.

1. More routine tasks, such as reporting and design release, will be automated. In the ultimate system, instead of machined part drawings sent to the shop, the design group data will be input for numerically controlled machines.
2. Initially, more time will have to be spent to define tasks and interfaces so that rework is minimized with a net reduction in time and money to do the jobs.
3. It will be advantageous to stress consistency (i.e., in the data base, units, coordinates and input-output requirements of related operational modules).
4. Engineers from one or more disciplines will be working side by side in the interactive graphics mode (i.e., aerodynamics and mass properties engineers to perform a sizing study, or stress and structures engineers to define the best carry-through structure for a wing-fuselage combination) to obtain usable answers much more quickly.
5. Engineers will have to get used to performing multiple assignments since IPAD will relieve them of time required to perform routine work.
6. Because IPAD is a total design process it will give greater visibility to both the project manager and the engineers; as a consequence IPAD will require the project manager to think more closely in disciplinary group terms and to use faster-response management techniques and it will also require the disciplinary engineer to think more closely in project-manager terms.

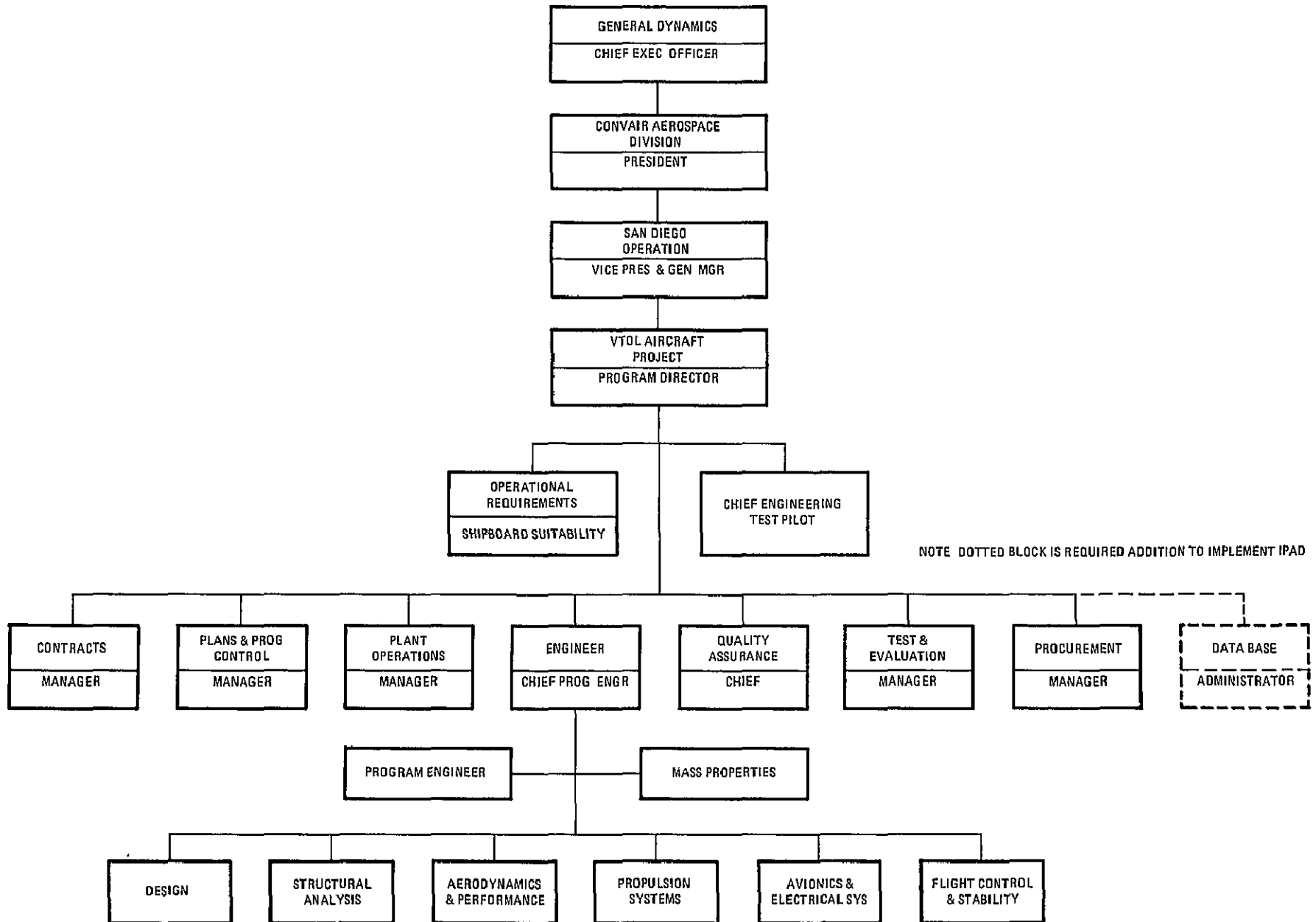


Figure 5-4. VTOL Aircraft Project Organization

These remarks indicate how any company can implement IPAD on a going project. It is envisioned that IPAD will be initially implemented to perform engineering and design tasks only for the conceptual design and preliminary design phases to show that it performs "as advertised", and to lay the groundwork for its full implementation in all disciplines such that all project tasks, including those for the detail design phase, can be performed.

5.2 Change in Design Organizations to Suit IPAD

The changes in design organizations to suit IPAD will be considered in two time frames, namely:

1. Near future.
2. Distant future.

Near future organization changes to exploit IPAD amount to using the "fly before you buy concept." The material discussed heretofore in this section showed how existing functional and project organizations with "a little bending" could be used to implement an initial IPAD system for the conceptual design and preliminary design phases, and to show that it performs "as advertised." It is seen that the change to existing organizations is extremely modest and the changes in existing procedures and practices are also modest. Thus, an initial IPAD "makes full use of existing organizations and design processes that work" and superimposes on them relatively modest changes in order to remove some of the deficiencies in these design processes and result in improved performance, decreased costs, and shorter schedules to complete the tasks.

Once the initial IPAD is developed and performs as specified, its future development will be evolutionary. Consequently, changes in design organizations to further exploit IPAD in the distant future cannot be precisely foreseen and defined at this time. However, conjectures on what some of these distant future changes might be are:

1. Discipline groups in both functional and project groups may disappear in some companies and be replaced with "system groups." For example, the current discipline groups called "Materials, Structural Dynamics, Stress, Mass Properties, and Structural Design" may be replaced by a single "Structural System Group" wherein a smaller number of interdisciplinary engineers and specialists perform the total design task using IPAD.
2. Certain operations, with corresponding change in organizational structure, may be eliminated. For example, some detail drawings and loft boards may be eliminated by having the designer through IPAD generate,

store, check, and transmit design data directly to a numerically controlled machine.

3. Because of the integrated team effort required by IPAD, some levels of supervision may be completely eliminated. Here the question will be asked "Just what does each supervisor contribute towards a specific design task?"
4. With expanded use of IPAD, economic considerations will have a hand in shaping the organization. For example, the administrative function called "Planning and Control" may be implemented in the group where the work is actually done.
5. As IPAD expands, changes in organization may come about to prevent unqualified people from misusing data and modules between various disciplines.
6. A given discipline group or a given system group may be asked to perform a new task that it had not done before and/or relinquish others that it had done heretofore. An example is the "parts count" task assigned to the Mass Properties group at Convair for use in estimating manufacturing costs.
7. Each company using IPAD will try to learn from its competitors and implement profitable changes in organization.
8. The training of the people in the organization will have to change so that they all know how to exploit the capabilities provided by an IPAD system.
9. As IPAD expands there may be a need for an individual in the project organization to ensure that "the user" controls IPAD and not vice versa, and that IPAD is used properly and with common sense.

5.3 Answers to NASA RFP Questions

5.3.1 What are the ramifications of traditional company design organizations and procedures relative to the acceptance and utility of IPAD? - Implementation of IPAD requires that top-level management recognize the need for it and specify that it shall be implemented on the company's programs. Initial implementation of IPAD on a going project in engineering for only the conceptual design and preliminary design phases would require extremely modest changes to current organizations and modest changes to existing design procedures and practices.

5.3.2 How will company design organizations likely change to use IPAD most effectively? - Near-term changes in organization for an initial IPAD would be very modest and relate mostly to project organizations. These changes include the addition of a Data Base Administrator, the use of automated management techniques to interface with quicker engineering evaluation cycles, and the training of project personnel in the use of IPAD. Further expansion of IPAD would be evolutionary, so that distant-term changes to organizational structure cannot be predicted accurately; however, conjectures on some possible distant-term changes include the possibility of creating "system groups" to replace several closely related disciplinary groups, the elimination of detail drawing and loft boards, the elimination of certain levels of supervision, and the shifting of task responsibilities from one group to another.

5.3.3 (Added by contractor, since it has been raised a number of times during the IPAD study) What happens if IPAD is not implemented? - It is believed that IPAD-type systems will evolve very slowly to satisfy individual company/agency needs and without proper integration. A manifold duplication of individual company efforts will result in extra costs, which can be avoided by an integrated approach.

6 IPAD SPIN-OFF ASSESSMENT (PHASE II, TASK 8)

In Section 4 of this volume it was pointed out that some major deficiencies exist in the current aerospace vehicle design process, and that IPAD could help correct or mitigate these deficiencies because IPAD is suited to solving problems where voluminous data is generated and manipulated.

IPAD spin-off in all non-aerospace fields, either as conceived for aerospace vehicles in this study or suitable variations thereof, will also correct or mitigate deficiencies in the solution of technical and business problems where voluminous data is generated and manipulated (i. e., where the work is characterized as an integrated effort of many sophisticated disciplines). In these other fields, just as in the aerospace field, IPAD will provide a fast response interactive environment to enhance the creative thinking done by a team of people having the responsibility for planning, problem solving, controlling, and managing a specific project or investigation.

The approach to the use of IPAD in non-aerospace fields should be similar to the approach used in this study pertaining to aerospace vehicles, namely define the current process, identify the deficiencies, and design and develop an IPAD-type system to correct or mitigate these deficiencies. Since the Hardware/Software complex required for IPAD can be used in all fields of sciences and engineering, it has to be developed only once and non-aerospace field teams must provide only their own automated capability to solve the problems peculiar to their disciplines and field.

The following sections identify potential spin-offs in greater depth. These spin-offs are seen to take place in both near-term and far-term time frames.

6.1 Spin-Off in Technical Non-Aerospace Fields

Since aerospace space vehicles are essentially a mode of transportation (i. e., even missiles are in this category since their function is to carry a payload to a destination) it is expected that the first near-term spin-offs in the use of IPAD will be in other modes of transportation. These are:

1. Marine transportation (i. e., water surface vehicles and underwater vehicles including underwater weapons like torpedoes and missiles).
2. Ground transportation.

- a. Automotive engineering.
- b. Mass transportation engineering (ground and underground).

Potential spin-offs in other technical non-aerospace fields are immense and will be limited only by the imagination of people involved in the design, manufacturing, and service operation of hardware. Illustrative examples of spin-offs in other fields where IPAD-type systems would be very beneficial to handle the voluminous data generated and manipulated are:

- 1. Communication.
 - a. Design and manufacture of hardware.
 - b. Traffic Control (i.e., air, water, and land vehicles).
 - c. Message control (i.e., sound, visual, and written).
- 2. Electronics (i.e., design and manufacture of hardware).

Some segments of the electronic manufacturing industry already employ IPAD-type systems where the electronic design engineer does his total job at interactive terminals. He performs circuit analysis, selects components, designs the circuit and lays out the circuit board, and these outputs are automatically retrieved and used in automated manufacturing, documentation, and administrative activities.

- 3. Civil Engineering.
 - a. Design and analysis of large structures (i.e., buildings, bridges, towers and dams).
 - b. Surveying (i.e., contour mapping, and improving accuracy of measurement of distances between two points that are very far apart, such as between two continents).
 - c. Highway and freeway layouts (i.e., layout studies usually made could be supplemented by studies to improve safety. At the present time the highways and the vehicles using them are each essentially designed as separate entities).
 - d. Pollution containment and control (i.e., data management for processing sensor data).

- e. Hydrology (i.e., flood control and drainage analysis).
 - f. Airport design (i.e., study improvements in service and safety from a total system standpoint).
4. Nuclear Engineering.
- a. Design and manufacture of reactors.
 - b. Monitoring of reactor service operation for safety.
 - c. Automatic data reduction when neutron sources are used for chemical analysis.
5. Mining Engineering.
- a. Oil exploration.
 - b. Mineral exploration.
6. Chemistry (i.e., automatic chemical analysis and identification, particularly when the analysis has to be made repeatedly as in air and water pollution control).
7. Meteorology (i.e., real time global weather prediction).
8. Mechanical Engineering.
- a. Design and manufacture of machined-part hardware (i.e., go from an interactive terminal to a machine shop, by-passing the drawing cycle). A coming new area is the design and manufacture of devices to aid handicapped people.
 - b. Minimize rework and scrap through Computer Aided Manufacturing (CAM).
9. Radio Astronomy (i.e., determine the physical dimensions of objects in space from intercontinental interferometers; this involves the processing and correlation of voluminous data).
10. Architecture (i.e., rapid pictorial and configuration displays).

11. Seismology (i.e., use of magnetometers in near earth satellites for advance warning of earthquakes).
12. Literature Retrieving (i.e., use IPAD-type techniques to help technical personnel keep abreast of the "literature explosion").

It is noted that IPAD is ideally suited to solve interdisciplinary problems that have heretofore been looked at only in a fragmented manner. For example, the pollution problem has many facets and, in addition to the pollution sensing and mitigating devices, it involves and must consider sociological, transportation, manufacturing, communication, legal, and policing problems in order to obtain an acceptable integrated solution. Similarly, the energy problem must consider existing forms of energy, available supply and location, consumption locations and rates, rationing, and development of new forms of energy to obtain a satisfactory integrated solution.

In addition to the spin-offs mentioned above, two others will automatically occur with IPAD type systems; namely,

1. Design, development, and manufacturing of improved IPAD-type systems and computer hardware and software
2. Training in the use of IPAD-type systems will increase in the industry, universities, and government agencies; this will make engineers, scientists, and managers more versatile.

6.2 Spin-Off in Business Non-Aerospace Fields

Just as in the case of the technical fields, IPAD-type systems will be beneficial to handle business problems in non-aerospace fields where voluminous data is generated and manipulated. A very important aspect of the application of IPAD-type systems to business problems is that it will give the manager much greater visibility much more rapidly than in the past. The end result of application of IPAD-type systems to business problems will be a combination of one or more of the following:

1. Improved performance.
2. Decreased costs.
3. Meeting short time schedules imposed by the Customer.

Illustrative examples of both near term and far term spin-offs in the use of IPAD-type systems in business non-aerospace fields and areas are:

1. Maintenance Management.

- a. In cases where many pieces of hardware are involved, keep records of the problems and fixes. From these records, define inspection methods and intervals; define preventive maintenance schedules and changes in design criteria; and design to improve the product and make it more cost effective over its service life. In most situations there is a paucity of factual maintenance records.
- b. Both industry and government have been faced with the problem of maintenance costs that over the life time of some expensive products have been higher than the original cost of the product. Systematic integrated analyses of these products should result in a marked reduction of maintenance costs to a small percentage of the original cost. This analysis would also be very effective in standardization; it would also point the direction of needed research. In our system of competing requirements for funds, attacking this problem in a vigorous integrated manner should be given very high priority.

2. Inventory Control. In cases where an organization has many products sold or used in many stores or departments and in many locations; rapid current status displays and future needs are mandatory for efficient low cost operation.
3. Conservation of Energy. To make systematic and integrated analyses of the way energy--electrical, compressed air, hydraulic, furnaces, etc.--is used with the view of conserving energy, and saving costs without reducing efficiency.
4. Legal Field. Use an IPAD-type system to quickly retrieve legal literature, particularly pertinent court rulings and decisions on cases.
5. Econometrics. Presently there are some elegant modeling techniques being used, in disparate disciplines, for getting an insight into the future effect of today's economic policies. For example, the Wharton School national economic model is used by many businesses. Significant benefits would accrue through integration of models created by various disciplines. The Environment Protection Agency could concentrate on modeling the relationships between environmental conditions

and the economy. The Department of Agriculture models food and fiber production, and should be interacting with the best weather predictions available from weather experts, whose predictions should be influenced by the amount of particulate matter in the atmosphere as predicted by other agencies. The State Department contributes relationships between our economy and the world economy. Natural resources use and remaining reserves in this country and abroad, need to be modeled for their effect on resource prices and the balance of payments. The potential of developing integrated models with multidisciplinary interfaces is obvious. Each specialty will be charged with continually updating its own modeling and evaluation capabilities so that the total model represents the best planning tool for everybody.

6. **Marketing Prediction.** To develop models and display results of marketing studies pertaining to goods, services, and entertainment.
7. **Hospital Administration.** Economy in the delivery of health services will foster more integrated application of computer technology than is presently achieved. IPAD-type systems can improve the use of medical data banks and statistics to help in quickly making diagnoses and prognoses of critically ill patients. Checking in patients, looking up the latest health insurance plan rules applicable, assuring proper billing of pharmaceutical services, and administration of medication are some of the multidisciplinary tasks that can be effectively integrated.
8. **Education.** From the point of view of transmitting knowledge, the teaching function is one of communications and systematic exposure of the students to "theory, examples, practice, historical cases, etc." associated with the subject at hand. Imbedded in this process is a continuous need to "visualize" examples, cause-effect relationships, end results, etc., which typically consumes a large percentage of the year or semester hours (both classroom and homework) available for each subject. The assembling of large multi-subject banks of interactive programs with pertinent examples, historical cases, etc. and organized on a subject basis, could provide a teacher-directed interactive system with the following features:
 - a. The shorter response times (in comparison to hand operations) afforded by an automated interactive system will permit either to:
 - (1) Expose the students to the same amount of teaching material in a shorter time span, or

- (2) During the same number of "subject hours" the students will be exposed to more example cases, better illustration of cause-effect relationships, more involvement. Furthermore, most of the functions of "homework" could be brought back to the classroom where the student can benefit from the teacher's coaching.
 - b. The teaching process would be accelerated mainly by removing most of the time-consuming drudgery for both teachers and students.
 - c. The students (both undergraduate and graduate) could be exposed to many real problems encountered in professional life (i.e. nature of the problem, how it was attacked and finally solved). IPAD is an excellent way to provide this capability.
 - d. The cost of higher education per professional would be reduced because of shorter educational cycles.
 - e. The capital investments per educational facility may go up to account for computer and special interactive equipment.
 - f. Graduating students will have a more comprehensive preparation for professional life.
9. Streamlining Procedures and Reducing Paperwork.
- a. IPAD-type thinking will require people in business areas to streamline procedures and reduce paperwork. Let us consider two examples to vividly illustrate this point. First assume a person desires to give someone a ten dollar gift; he accomplishes this by simply pulling out his wallet, taking out ten dollars and handing it to the person--this is a neat, clean transaction that is accomplished in final or closed form in a brief instant; IPAD-type thinking should not be used here in this very simple and efficient operation. Next assume a person desires to give someone an insurance policy as a gift; here are the factors he must consider in the transaction:
 - (1) Who is the donee and what is his age?
 - (2) Who will be the guardian of his estate if donee is a minor?

- (3) What is the relationship between the donor and the donee, and in what State was the gift made?
- (4) What is the name and address of the insurer, the number of the policy, when purchased, and its face value?
- (5) What is the premium, when is it due, who pays it, and what is the source of these funds?
- (6) What is the date of the gift and does this date have any special significance (i.e., birthday or Christmas gift)?
- (7) What is the interpolated terminal reserve value of the policy, and what is the relationship of this gift to other gifts made by the donor during his lifetime?
- (8) Is the gift one of present interest or one of future interest and what evidence does the donor have that it is one or the other?
- (9) If the gift is to be in trust, is the trust funded, and what is the employer identification number?
- (10) Execution of absolute assignment forms (it is noted that each insurance company uses a different form with different size, color, and words).
- (11) Compliance with state gift tax laws and execution of gift tax forms (it is noted that considerable differences exist between gift tax laws in the various States).
- (12) Compliance with federal gift tax laws and execution of gift tax forms (it is noted that these laws and forms differ from those in the States); one also has to contend with conflicts in tax court decisions where two different tax courts arrive at conflicting decisions based on essentially the same facts.

Thus the procedures and forms used in the United States to make a gift of an insurance policy literally consist of a very large encyclopedia. In essence, this is similar to a technical interdisciplinary problem, and an IPAD-type approach would show how

to streamline the procedures and markedly reduce the paperwork for accomplishing such a transfer.

- b. Similarly, IPAD-type thinking and procedures applied to other businesses (e.g., banks, post office, hospitals, real estate, brokerage houses, welfare and tax agencies) will point the direction towards streamlining procedures and reducing paperwork.

Since IPAD is a new tool, the extent of its full broad utilization will become evident in an evolutionary manner.

6.3 Answers to NASA's RFP Questions

This section presents answers to questions posed by the NASA in relation to Task 8 and questions added by the contractor. In the following subsections, each question is quoted verbatim from the RFP, followed by the answer.

6.3.1 Will experience gained in the implementation of a system like this open the way to the creation of similar interdisciplinary systems in non-aerospace fields? - Yes. Although there are exceptions, most non-aerospace fields deal with design and operational problems which are interdisciplinary. These problems are typically broken down and solved as single-discipline subproblems. The experience to be gained with IPAD in multidisciplinary aerospace field applications will be an example for other technical and non-technical fields. The IPAD system framework developed for the aerospace industry can be used directly by many other technical/scientific fields utilizing their own Operational Modules.

6.3.2 What are these fields? - Some of the technical and business fields are noted in Sections 6.1 and 6.2 of this volume.

6.3.3 (Added by contractor). How rapidly can one expect to have IPAD-type systems applied to non-aerospace fields? - Non-aerospace industries that handle large volumes of data and that can exploit automation and an interactive environment will be immediate beneficiaries of an IPAD system. Some government agencies, such as the U. S. Navy, are presently developing IPAD-like systems to fit their specific needs in ship design. Foreign and domestic car manufacturers are using various features of IPAD-type systems that have been developed within the constraints of present equipment and software. The need to reduce costs will be the stimulant to entice "late starters" to adopt IPAD-type systems to streamline their operations.

REFERENCES

1. Milliken, J. G.; Morrison, E. J.: Management Methods from Aerospace. Harvard Business Review, Vol. 51, No. 2, March-April 1973.
2. Anon: Computer Program Manual. National Aeronautics and Space Administration, Langley Research Center, Volume II, Section 3.2.1, November 1967.
3. Chmura, J. T.; Johnson, B. V.; Smith, J. L.: A Data Base Management System Modelled on the CODASYL Proposals. Australian Joint Computer Conference, 1972.
4. Telecon between Carol Horwitz, of the University of Wisconsin Medical Research Center's Laboratory Computer Facility, and Mike Hurley of Convair Aerospace Division of General Dynamics - RE: their ISIS System - 16 March 1973. (Follow-up communique, 21 March 1973, contained four preliminary documents supplying selective details of their approach.)
5. "Panel Reports for DOD/Industry Symposium on Computer-Aided Design, and Computer-Aided Manufacturing/Numerical Control," held at Davenport, Iowa, Rock Island Army Arsenal, 14-17 October 1969, Panel No. 3, State of the Art in CAD, by Roy M. Salzman of A. D. Little, Inc.
6. Anon: STOL Tactical Aircraft Investigation, General Dynamics/Convair Aerospace Division Report GD/CA-DHG-73-001, Vol. I Configuration Definition, March 1973.
7. Anon: Convair Model 200 Navy V/STOL Fighter Attack, General Dynamics/Convair Aerospace Division Report GDC PSA 72-1004, Vol. VII, Structures, June 30, 1972 (Confidential).
8. Anon: Research and Applications Module (RAM) Phase B Study; Technical Data Document, General Dynamics/Convair Aerospace Division Report No. GDCA-DDA-72-006, Volume III, Preliminary Design, Appendix F, Application of Advanced Composite Materials to Free-Flying Ram, dated May 12, 1972.
9. Bonine, K. C., Interactive Computer Graphics for Computer Aided Design, General Dynamics/Convair Aerospace Division Report No. GDCA-ERR-1727, December 1972.

☆ U. S. GOVERNMENT PRINTING OFFICE: 737-508/87 REGION NO. 3-II.